(1)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GE/ENG/88D-14

MULTIVARIABLE CONTROL LAW DESIGN FOR THE CONTROL RECONFIGURABLE COMBAT AIRCRAFT (CRCA)

THESIS

Daryl Hammond
Captain, USAF

AFIT/GE/ENG/88D-14

AFIT/GE/ENG/88D-14

MULTIVARIABLE CONTROL LAW DESIGN FOR THE CONTROL RECONFIGURABLE COMBAT AIRCRAFT (CRCA)

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Daryl Hammond, B.S.
Captain, USAF

December, 1988

Approved for public release; distribution unlimited

Acknowledgments

Sincerest thanks are given to my thesis advisor Professor John J. D'Azzo. Without his support, guidance, and confidence in my ability this thesis would not have been possible. His technical expertise, tireless reviews, and dedication make him truly an outstanding engineer and person. Gratitude is also in order for Dr. Robert Fontana and Major Phil Amburn for providing insightful conversations and helping me keep the research effort in proper perspective. I also wish to thank Mr. Curtis Clark and Captain Robert Eslinger of the Flight Dynamics Laboratory. Mr. Clark was always patient and willing to answer a technical question on the aircraft and his professional attitude, knowledge, and devotion to his work has helped bridge the gap between theory and practical control law design. Captain Eslinger not only sponsored this research effort, but devoted valuable time in making comments and suggestions on the thesis write up.

Deepest and heartfelt thanks are expressed to my wonderful wife [redacted] daughters, [redacted] but somehow these thanks don't seem enough to compensate for the hardships that they have endured. [redacted] continually provided love and encouragement while the children were always there with a hug and kiss when I was particularly "grouchy." Finally, thanks are given to the Lord Jesus Christ my Savior who has provided me the strength and wisdom to survive this program.

Daryl Hammond

Table of Contents

List of Figures

Figure		Page

Figure | Page

Figure	Page

Figure Page

Figure	Page

List of Tables

AFIT/GE/ENG/88D-14

Abstract

Typically, control law analysis and design for an aircraft include separating the longitudinal and lateral equations of motion and designing control laws for each separate motion. The simplifying assumptions are often valid and do not adversely affect the analysis and design when aerodynamic cross-coupling is minimal. The Control Reconfigurable Combat Aircraft (CRCA) design includes an all-flying canard with 30 degrees of dihedral angle which prevents the normal separation of lateral and longitudinal equations because of high aerodynamic cross-coupling. Consequently, developing a satisfactory controller for all aircraft motion must include all of the control surfaces and is more complicated.

The multivariable control law design used in this thesis incorporates the high-gain error-actuated Proportional plus Integral (PI) controller developed by Professor Brian Porter of the University of Salford, England. Control law development and simulation are performed using the computer aided design program called MATRIX$_x$. Two successful fixed gain controller design methods and an adaptive controller design are demonstrated.

The three control surfaces on each wing are operated together, so they are treated in this thesis as one control effector. Thus, the five CRCA control inputs for this design consist of two canards, left trailing edge flaperon, right trailing edge flaperon, and rudder. The aircraft dynamics are linearized about three flight conditions. Fixed gain PI controllers are designed at each flight condition for both the healthy aircraft and with a failed left canard and left trailing edge flaperon. The simulation indicates that the controller is very robust and output responses are fully satisfactory.

An adaptive controller design, using a recursive least squares (RLS) parameter estimation algorithm, is developed for a self-tuning control system. When a left

trailing edge flaperon failure occurs at the nominal flight condition, a new plant model is computed and the PI control law is revised accordingly. The ability of the controller to estimate the new plant parameters and compensate for the failed control surface is exceptional. Finally, a fixed gain Proportional plus Integral plus Derivative (PID) controller is designed for the nominal flight condition. The results are satisfactory. The design needs to be extended to the remaining flight conditions.

MULTIVARIABLE CONTROL LAW DESIGN FOR THE CONTROL RECONFIGURABLE COMBAT AIRCRAFT (CRCA)

I. Introduction

1.1 Background

Countering state-of-the-art advances in enemy air and ground threats places strenuous and seemingly impossible demands on the modern day fighter aircraft. In aerial combat, the pilot requires agility and maneuverability from his airplane, even at high speeds and under high loading conditions. For pin-point bombing and target tracking, precise heading and flight path control is essential. To complicate matters, aircraft controllability must be maintained even with control surface failure or damage, a real possibility in combat. In a digital fly-by-wire flight control system (FCS), the FCS is critical for safe flight and becomes the only link between the pilot and the control surfaces. The FCS computer has complete control and optimally deflects available control surfaces for aircraft positioning and stability during these control surface failures.

An early example of a control system design was one used to control aircraft heading, altitude, and speed. In 1933, Wiley Post used the Sperry pneumatic-hydraulic Gyropilot to accomplish his around the world flight in less than eight days. In 1947, an Air Force C-47 made a completely *automatic* transatlantic flight, including takeoff and landing using autopilot control [4, 1-4]. However, these simple control systems lack the sophistication needed to manage the additional control complications introduced by high performance aircraft design. In order to meet enhanced mission capabilities for handling and response, aircraft designers no longer

restrict themselves to providing statically stable aircraft. If an aircraft is slightly unstable, a disturbance from an equilibrium condition will be performed much quicker. This is desirable in rapid maneuvering, however the pilot may not be able to keep up with the fast changes without help from the flight control system. By incorporating full-authority digital fly-by-wire flight control systems using fast onboard microprocessors, these aircraft can be stabilized and high performance control configured vehicle (CCV) maneuvers such as pitch pointing, lateral and vertical translation, and high gravitational loading, "g", tracking tasks are possible. These maneuvers require special use of all control surfaces, including independent operation of right and left elevator surfaces, flaps, or other special purpose controls such as thrust vectoring. Pilots are often preoccupied with navigation or weapons delivery work and rely on the FCS to manage all control surfaces in their proper proportions, especially if control surface effectiveness or complete failure occurs [9].

Catastrophic consequences can result when a control surface malfunctions or is damaged by enemy fire. Maintaining aircraft control, pilot safety, and combat effectiveness is of primary concern and places exceptional demands on the pilot and control system. Sensors need to detect control surface damage and special purpose software algorithms must isolate the damage and direct the information to the control system computer for rapid reconfiguration of the aircraft control law. In order to maintain aircraft stability, control surfaces designed primarily for longitudinal control may be used to generate lateral control forces and moments [9, 3]. For instance, differential elevator deflections can be used to produce a rolling moment should aileron or canard movement be unobtainable.

The benefits from fault identification and reconfiguration of an aircraft are high in terms of FCS vulnerability reduction, but certainly are not without cost. Control law design is complicated by the performance demands for maneuverability and reconfiguration capability. The control engineer is faced with implementing

numerous complex feedback control loops where control surface positions, velocities, and accelerations are recombined to insure stability and performance [2, 3]. Numerous iterations and "tweeking" of the feedback loop gains are required for the design. For complex multiple-input and multiple-output (MIMO) system design techniques which use output feedback, the process may be time consuming. Consequently, it is desirable to implement a control law that can be easily designed in a few steps.

1.2 Problem Statement

The objective of this thesis is to develop a parameter-adaptive controller for the Control Reconfigurable Combat Aircraft (CRCA) that has satisfactory performance for a failed, or partially failed, left flaperon in the Air Combat Maneuver (ACM) Entry flight condition. This objective is accomplished by first developing continuous and discrete time domain fixed gain controllers for operation at other points in the CRCA flight envelope, including control surface failure conditions, to provide a design base for the adaptive controller implementation. Stability and performance of the crippled system are then evaluated and compared to the healthy aircraft. The performance criteria used in system evaluation include such items as overshoot in commanded inputs, steady-state error in output responses, and control surface deflection rates and limits.

1.3 Summary of Current Knowledge

The conventional or classical control theory is extremely effective when the aircraft is described by a set of single-input and single-output (SISO) control relationships [8]. Root locus and frequency domain analysis give good insight into the control law implementation, but are less effective for MIMO systems that are encountered with modern aircraft design. Modern control techniques include eigenstructure assignment and optimal control. Both techniques use combinations

of state variable and output feedback. However, output feedback is often the preferred method of control law design because of the difficulty in obtaining, or estimating, all the states necessary for a state feedback system [8]. A high gain output feedback multivariable design technique, developed by Professor Brian Porter of the University of Salford, England, shows great promise in the design of modern control law systems for high performance aircraft [17,18,19,22].

1.3.1 Multivariable Control Theory The design technique implements a Proportional plus Integral (PI) controller by using singular perturbation theory based on the mathematics of linear algebra [6,9,17,19,22,20]. The high-gain used in the design drives the closed-loop response of the system to two distinct sets of modes, referred to as fast and slow modes [8, 660-661]. This asymptotic behavior of the modes of the system essentially decouples the outputs, where the fast modes primarily give the system a fast tracking capability as the gain of the system is increased. The slow modes are asymptotically uncontrollable and unobservable and therefore make only small contributions to the output, thus the output is dominated by the fast modes [8]. Compensator gain values are chosen by the designer, based on engineering insight and experience, to decouple the system outputs and yield a good response. However, the controller gains are selected for a given system configuration, and for large variations in plant parameters these values may not give satisfactory performance. In such cases an adaptive or variable gain controller is needed.

1.3.2 Adaptive Control Blakelock defined a self-adaptive system as one which has the capability of changing controller parameters through an internal process of measurement, evaluation, and adjustment to a changing environment [4, 199]. As flight conditions change due to altitude, airspeed, angle of attack (AOA), and structural effects, the performance of the Proportional plus Integral fixed gain controller design may not be sufficient to compensate for the changes.

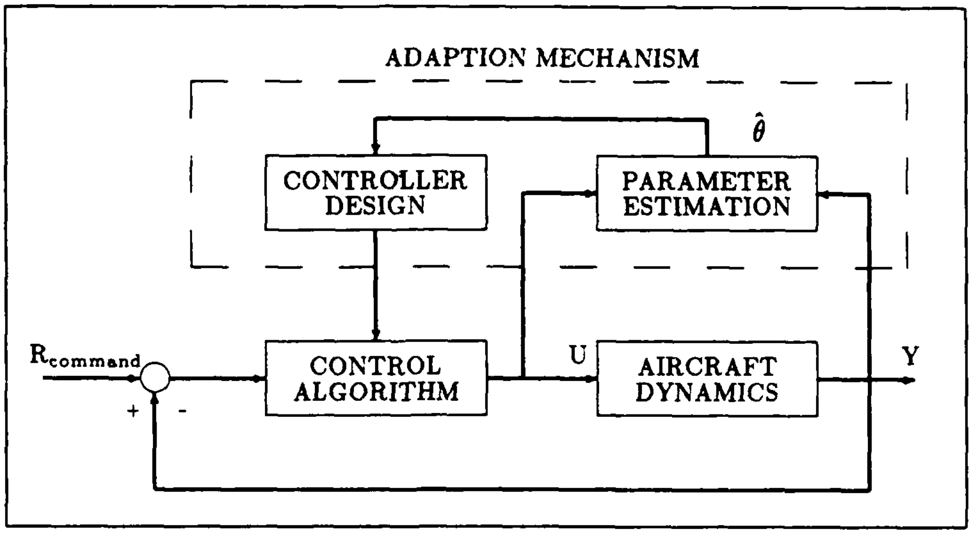


Figure 1.1. Self-Tuning Regulator

The control system can be stabilized by gain scheduling or self-tuning controllers [1]. Because aircraft transfer functions change in proportion to airspeed, angle of attack (AOA), and altitude, these parameters are often used to gain schedule [5, 199]. With the use of high speed digital and analog computers in fighter aircraft, gain changing is made quickly and smoothly. Computer memory hardware is small, light weight, and inexpensive, thus allowing storage of numerous controller gains. In addition, there are self-tuning compensators that contain estimating schemes for optimal gain scheduling. Figure 1.1 illustrates the technique used to calculate gains based on parameter estimation [27]. These estimating schemes include recursive least squares (RLS), recursive maximum likelihood, and stochastic approximation [1]. Porter and Fripp demonstrated that the adaptive system could be applied to a multivariable plant with the PI controller in a reconfiguration design [18]. Their results showed the effectiveness of the adaptive controller to "re-tune" the control law for an actuator failure. [18].

1.4 Assumptions

For this thesis the linearized equations of motion for the CRCA are used and the following basic assumptions are made [2,7,9].

1. The aircraft is a rigid body, and mass is constant.
2. The earth's surface is an inertial reference frame.
3. The atmosphere is assumed fixed with respect to the earth.
4. Linearization about an operating condition is acceptable for point designs.
5. Aerodynamics are constant for altitude and Mach number.

The rigid body assumption (one point on the aircraft remains fixed with respect to every other point), is made in order to reduce the complexity of the aircraft linearized equations of motion. Therefore, bending modes and structural flexibility and interaction between control surfaces are not considered in this study. Although the mass of the aircraft is constantly changing due to fuel consumption, the period of time used in the simulation, 1 - 10 seconds, is considered negligible and aircraft mass is considered constant.

The earth centered reference frame assumption greatly reduces the size and difficulty in deriving the equations of motion. Valid results are possible because the maneuvers performed are short in duration compared to the rotation time of the earth, and the sensors used to detect rates are not accurate enough to detect the earth's rotational components during the performed maneuver [2, 8]. The effects of changes in atmospheric conditions such as high winds, wind shear, and gusts are beyond the scope of this thesis.

In order to develop control laws that will be effective over a range of operating conditions, a large number of designs within the flight envelope need to be investigated. This thesis considers only three flight conditions: Air Combat Maneuver (ACM) Entry, ACM Exit, and Terrain Following/Terrain Avoidance (TF/TA) as

a cross section of operating conditions within the flight envelope. The aerodynamic data has been provided by the Flight Dynamics Laboratory. ACM Entry corresponds to the initial maneuvering to position the aircraft for air-to-air missile deployment. Aircraft speed is usually in the transonic region at medium altitudes, 20,000 to 40,000 feet [26, C-1]. For this design a flight condition of 0.9 Mach and 30,000 feet is chosen. ACM Exit commonly refers to a flight condition associated with the end of an air-to-air confrontation, that is, low aircraft speed, high angles of attack, and high gravitational "g" load factors on the aircraft. For this design a flight condition of 0.275 Mach at 10,000 feet and a gravitational loading of 3 g's is selected. TF/TA is a high speed and low altitude flight condition associated with Entry and Exit from an enemy territory. The combination of high speed and low altitude can potentially cause high load factors should rapid pitch changes be necessary. For this design a flight condition of 0.9 Mach at Sea Level is chosen.

The validity of analyzing an aircraft about a fixed flight condition is based on the small perturbation assumption when the non-linear equations of motion are linearized at an operating condition. That is, the derived aircraft model is valid only for small changes in aircraft configuration or deviations from the linearized "trim" point. Large changes in aircraft speed or altitude cause the small perturbation assumption to be invalid because aircraft stability coefficients change with variations from this trim.

1.5 Approach

The Control Reconfigurable Combat Aircraft (CRCA) is chosen for this study. The lateral and longitudinal equations cannot be decoupled for this aircraft, especially during the ACM Exit flight segment, commonly referred to a low $\bar{q}$ condition. Because of the effectiveness in pitch, roll and yaw, the dihedral canard control surfaces and aerodynamic design prevents the normal separation into lateral and longitudinal equations and complicates the controller design process.

Developing a satisfactory controller provides a challenge for the development of a high gain error-actuated controller design using the Porter method.

The aircraft consists of nine individually controlled control surfaces; two canards, four trailing edge flaps, two elevator surfaces, and a rudder. For this design the control surfaces for the trailing edge flaps and elevator are commanded simultaneously to yield an "effective" control surface of a left and right trailing edge surface. This reduces the number of controlled surface deflections to five: two canards, left trailing edge flaperon, right trailing edge flaperon, and rudder. The control law for the healthy aircraft is first designed using Porter's high-gain error-actuated controller method. A computer aided design program called MATRIX_x is used for the design process [15]. To demonstrate the robustness of the control law at each flight condition, a pitch pointing maneuver and a 45 degree coordinated turn are commanded. These maneuvers are chosen to demonstrate both the lateral and longitudinal control capability of the designed compensator. Since maintaining aircraft control is essential when surface failure occurs, the control law developed for the healthy aircraft is evaluated when a reduction of control surface effectiveness occurs. The single failure consists of a 50 percent loss of effectiveness of the left canard and 30 percent loss of effectiveness of the left flaperon. These control surfaces are prominent projections and highly vulnerable to damage during enemy confrontation.

The second part of this thesis investigates the design of a parameter-adaptive controller for the same aircraft. The commanded maneuver is a 45 degree coordinated turn at the ACM Entry flight condition. The introduction of a parameter-adaptive control design demonstrates an ability to update the PI control law gains when a partially failed left trailing edge flaperon control surface failure occurs. The technique uses a recursive least squares (RLS) identification algorithm to determine the updated step-response matrix when plant parameters change. According to Professor Porter, the ability of the controller to "re-tune" the control laws is

highly effective when failures occur [18].

1.6 Overview

The remaining material in this thesis is organized as follows: Chapter 2 contains a discussion of the CRCA aircraft model and plant dynamics. Chapter 3 provides the theory behind the PI control law implementation and the adaptive control algorithm and parameter estimation scheme. The PI controller design for the nominal flight condition, ACM Entry, is presented in Chapter 4. The results of the fixed gain controller are given in Chapter 5. The adaptive controller design and implementation is provided in Chapter 6. Conclusions and recommendations for future study are presented in Chapter 7. Appendix A contains a complete set of aircraft data, including state space and auto regressive moving average (ARMA) models. The theory for calculating the ARMA plant representation is covered in Appendix B. Appendix C contains the time history graphs of the plant parameter changes for the adaptive PI controller. The model-following Proportional plus Integral plus Derivative (PID) control law development and implementation for the nominal flight condition, ACM Entry, is presented in Appendix D. The study of the PID controller is not as extensive as the other method. Appendix E shows the complete state and output responses for the discrete and continuous time domain PI controllers designed for the nominal flight condition. The MATRIX_x simulation macros and RLS FORTRAN code is presented in Appendix F.

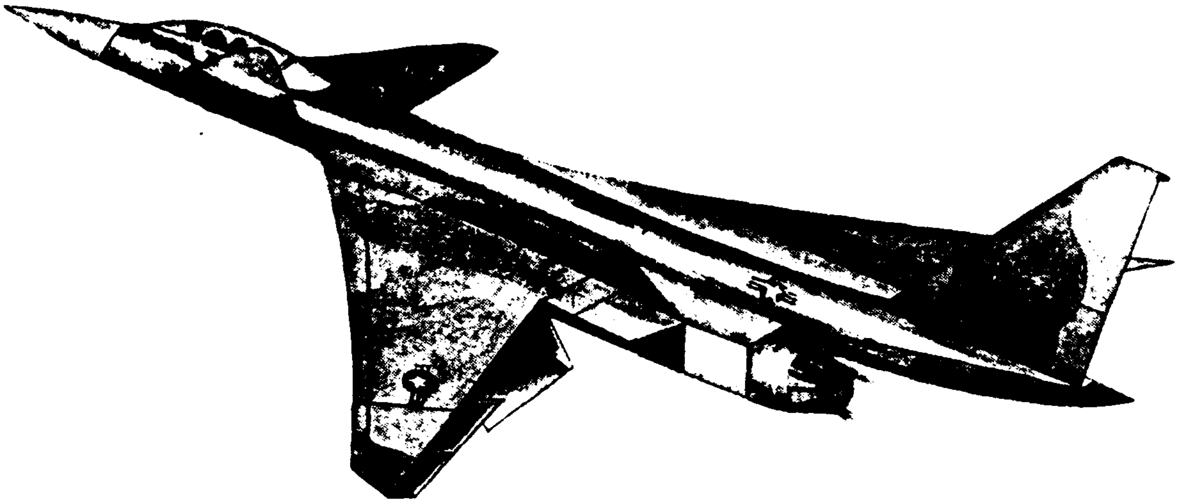
II. *Aircraft Description and Models*

2.1 Introduction

This chapter briefly describes the Control Reconfigurable Combat Aircraft, CRCA, and the aircraft models that are used in the control law analysis. The information for this section of the thesis is obtained from the Control Reconfigurable Combat Aircraft Development Phase 1, Research and Development Evaluation provided by the Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio [26].

2.2 Aircraft Description

A sketch of the Control Reconfigurable Combat Aircraft, CRCA, is shown in Figure 2.1. The aircraft consists of two trailing-edge flaperons and an elevator per wing, a vertical rudder, and right and left canards with 30 degrees of dihedral. The four flaperons and two elevators are primarily used in providing pitch and roll control, but also supplement the canards with independent movement when stability is needed should canard saturation occur during a maneuver. Control surface movements of the flaperons and elevators fall into two groups, symmetrical and differential. Symmetrical movement of the trailing edge surfaces enables longitudinal trim and airflow control, pitch control, and airfoil lift during landing and air combat maneuvers. Differential deflection of the elevators and flaperons permits precise lateral trim and roll control. The primary function of the vertical rudder surface is to provide directional trim and yaw control. Canard surfaces may also be operated symmetrically to produce pitching motion or differentially to produce yaw or roll moments. The 30 degrees of dihedral angle significantly increases the effect of the canards both in the differential and symmetric deflection categories, but the dihedral angle reduces directional stability. For the designs in this thesis, the trailing edge flaps and elevators are commanded simultaneously to yield "effec-



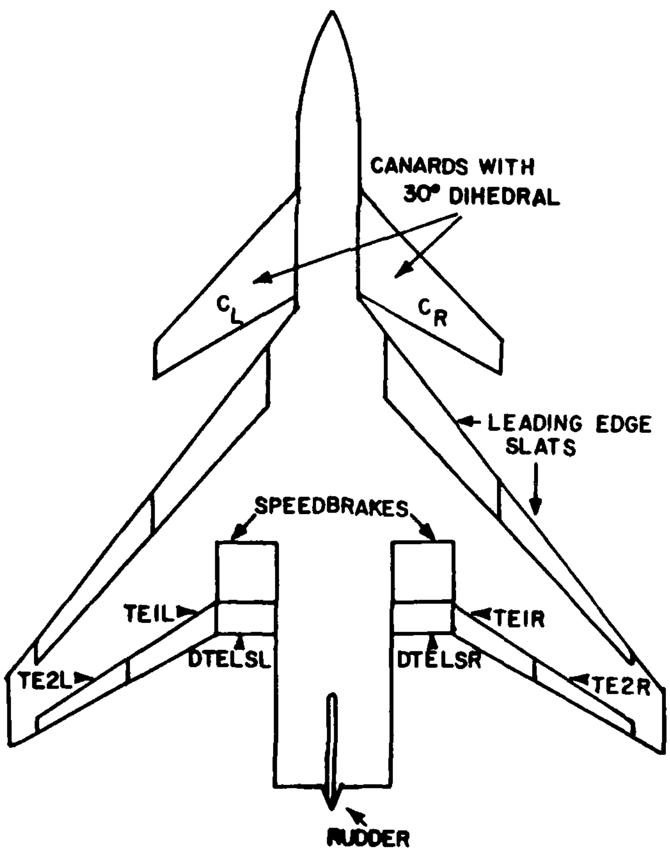


Figure 2.1. Combat Reconfigurable Control Aircraft (CRCA)

Table 2.1. Mission Segments

MISSION SEGMENT	ALTITUDE, FT	MACH NO.	LOAD FACTOR
STOL	1200	0.185	1g
TF/TA	SEA LEVEL	0.9	1g
ACM ENTRY	30,000	0.9	1g
ACM EXIT	10,000	0.275	3g

tive" left and right trailing edge control surfaces. This arrangement is consistent with existing control law design and provides preservation of laminar air flow over the wing during control surface movement. The number of controlled surface deflections are reduced to five: two forward canards, left trailing edge flaperon, right trailing edge flaperon, and rudder. Therefore, the control surface deflections are defined as follows:

$$\begin{aligned}
\delta_{cl} &= \text{left canard} \\
\delta_{cr} &= \text{right canard} \\
\delta_{tel} &= \text{left trailing edge flaperon} \\
\delta_{ter} &= \text{right trailing edge flaperon} \\
\delta_{rud} &= \text{rudder}
\end{aligned}$$

The CRCA flight control system is designed to perform satisfactorily in four mission segments, or flight conditions. Although design is at four set points, the control law is valid over a range of set points in terms of maneuvers. These flight conditions are chosen to feature the aircraft's control power during static and dynamic controllability analyses. Table 2.1 gives a brief description of each flight segment.

The first flight segment, Short Take-off and Landing (STOL), involves maneuvering at low speeds and is considered one of the most important flight segments. This capability allows for continued mission operations from damaged or non-ideal

airfields. Typically, the STOL landing requirements are defined as a maximum landing distance of 1500 feet and a maximum crosswind of 30 knots. Since aircraft configuration is different at this flight segment and often requires special gain scheduling, this thesis concentrates on a controller design for the remaining three flight conditions. This decision results in the demonstration of the robustness of the design technique in analysis of aircraft performance through the remaining flight conditions.

The second flight segment, Terrain-Following/Terrain-Avoidance (TF/TA), involves high speed flight within 200 feet of the terrain. This flight condition is normally associated with quick entry and return from enemy territory where low altitude and high speed reduce the risk of detection and elimination. However, the combination of high speed and low altitude can place severe stresses on the aircraft and pilot should rapid pitch maneuvering be necessary.

The third flight segment, Air Combat Maneuvering Entry (ACM Entry), involves considerable maneuvering normally associated with air-to-air weapons delivery or enemy aircraft evasion. Aircraft speeds are usually in the transonic region with confrontation altitudes between 10,000 and 30,000 feet.

The final flight segment, Air Combat Maneuvering Exit (ACM Exit), represents low aircraft energy at the termination of an air-to-air confrontation, that is, high angle of attack, low air speed, and high load factor. This flight segment involves the region of largest inertial and aerodynamic coupling and is the most difficult condition for controller design because of limited control power. Typical aircraft attitude includes a trimmed angle-of-attack of 30 degrees, bank angle of approximately 70 degrees, and sustained loading of 3 g's.

This thesis investigates the design of fixed gain proportional plus integral (PI) controllers for the ACM Entry, TF/TA, and ACM Exit flight segments. The adaptive controller design is implemented for the nominal flight condition, ACM Entry, and associated control law adaption for a 30 percent loss of effectiveness of

the left trailing edge.

2.3 Aircraft Models

2.3.1 State Space Model The CRCA aircraft state and output equations can be written in the form

$$\dot{x} = Ax(t) + Bu(t) \tag{2.1}$$

$$y(t) = Cx(t) \tag{2.2}$$

where,

A = the continuous time plant matrix (n x n)

B = the continuous time control matrix (n x m) with the rank of B = m

C = the continuous time output matrix (p x n)

x = the state variable vector with n states

u = the control input vector with m inputs

y = the output vector with p outputs

The state and output equations are obtained from the aircraft equations of motion for forces and moments acting on the aircraft. A Flight Dynamics Laboratory computer program for the CRCA was used to obtain the linearized equations of motion, stability coefficients, and the corresponding state space models. Table 2.2 illustrates the aircraft modeling information for the nominal flight condition, ACM Entry.

Table 2.2. Aircraft Data for ACM Entry

Flight Condition

M	=	.9	$\bar{q}$	=	356.3 lb/ft^2
Alt	=	30,000 ft	Thrust	=	6134 lb
V	=	895 ft/sec	DLFLP	=	0.0 deg

Aircraft Attitude

α	=	2.02 deg	θ	=	2.02 deg
β	=	0.00 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-5.09	TE1L	=	2.586
CR	=	-5.09	TE1R	=	2.586
DTE1SL	=	2.586	TE2L	=	2.586
DTE1SR	=	2.586	TE2R	=	2.586
RUDR	=	0.0			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0063 (deg^{-1})	C_{mq}	=	-3.4000 (rad^{-1})
$C_{n\alpha}$	=	.0761 (deg^{-1})	C_{nu}	=	-.0110 $(mach^{-1})$
$C_{a\alpha}$	=	-.0039 (deg^{-1})			

Lateral/Directional

$C_{y\beta}$	=	-.0067 (deg^{-1})	C_{np}	=	-.0290 (rad^{-1})
C_{lr}	=	.0610 (rad^{-1})	$C_{l\beta}$	=	.0014 (deg^{-1})
C_{lp}	=	-.2530 (rad^{-1})	C_{nr}	=	-.5270 (rad^{-1})
$C_{n\beta}$	=	.0012 (deg^{-1})			

Table 2.2 (continued)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00588	.00146	.00019	.00132	.00022	.00136
Right Canard	.00588	.00146	.00019	-.00132	-.00022	-.00136
L Obd Flaperon	-.00125	.00286	.00040	-.00043	.00053	.00006
L Inbd Flaperon	-.00165	.00443	.00061	-.00067	.00064	.00008
Left Elevator	-.00116	.00338	.00047	-.00051	.00032	.00006
Right Elevator	-.00116	.00338	.00047	.00051	-.00032	-.00006
R Inbd Flaperon	-.00165	.00443	.00061	.00067	-.00064	-.00008
R Obd Flaperon	-.00125	.00286	.00040	.00043	-.00058	-.00006
Rudder	0	0	0	.00255	.00033	-.00152

Table 2.3 illustrates the plant matrices with corresponding state variable vector x and control input vector u that are obtained from the linearization program for the ACM Entry flight condition. Aircraft models for the remaining flight conditions and corresponding program output are contained in Appendix A.

For this design the state matrices A and B of Table 2.3 are rearranged into a form more convenient for the Porter design technique. Recall that control surfaces for the trailing edge flaps and elevator are commanded simultaneously to yield "effective" left and right trailing edge control surfaces. The columns in the B matrix corresponding to the combination of control surfaces are summed together to yield the "effective" B matrix column value. Additionally, the redundant state, $\dot{\psi}$, is removed from the A matrix; thus the A matrix has full rank, $n = 8$. Also, the matrix is rearranged to move the kinematic equations to the first two rows of the A matrix. This results in an A and B matrix of the form illustrated in Table 2.4, with 8 states and 5 control inputs.

The output vector y, as defined in Equation 2.2, consists of outputs needed to command the desired maneuvers and is obtained from the output matrix, C. The output vector for this thesis is given by

$\mathbf{v}$ = forward velocity

β = side slip angle

θ = pitch angle

ϕ = bank angle

$\mathbf{r}$ = yaw rate

and the corresponding output equation

Table 2.3. Aircraft State Space Matrices for ACM Entry

$$A = \begin{bmatrix} -.0119 & -.0186 & -31.2350 & -32.1804 & .0000 & .0000 & .0000 & .0000 & .0000 \\ -.0324 & -1.0634 & 894.4548 & -1.0634 & .0000 & .0000 & .0000 & .0000 & .0000 \\ -.0002 & .0069 & -.6015 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0010 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0349 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0360 & -.0929 & .0349 & -.9994 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & -27.8066 & -2.0376 & .4913 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 2.4582 & -.0241 & -.4377 \end{bmatrix}$$

$$B = \begin{bmatrix} .0411 & .0411 & .1322 & .0866 & .1322 & .0866 & .1018 & .1018 & .0000 \\ -.3163 & -.3163 & -.9597 & -.6194 & -.9597 & -.6194 & -1.0183 & -1.0183 & .0000 \\ .1014 & .1014 & -.0284 & -.0215 & -.0284 & -.0215 & -.0200 & -.0200 & .0000 \\ .0003 & -.0003 & -.0002 & -.0001 & .0002 & .0001 & -.0001 & .0001 & .0006 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0010 \\ .0762 & -.0762 & .2219 & .2011 & -.2219 & -.2011 & .1109 & -.1109 & .1144 \\ .0486 & -.0486 & .0029 & .0021 & -.0029 & -.0021 & .0021 & -.0021 & -.0544 \end{bmatrix}$$

$$x = \begin{bmatrix} u & w & q & \theta & \psi & \phi & \beta & p & r \end{bmatrix}^T \tag{2.3}$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{te1l} & \delta_{te1r} & \delta_{te2r} & \delta_{dtelsl} & \delta_{dtelsr} & \delta_{rud} \end{bmatrix}^T \tag{2.4}$$

Table 2.4. Equivalent Aircraft A and B Matrices For ACM Entry

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0350 \\ -32.1804 & .0000 & -.0119 & -.0186 & -31.2350 & .0000 & .0000 & .0000 \\ -1.0634 & .0000 & -.0324 & -1.0634 & 894.4548 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0069 & -.6015 & .0000 & .0000 & .0000 \\ .0000 & .0360 & .0000 & .0000 & .0000 & -.0929 & .0349 & -.9994 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -27.8066 & -2.0376 & .4913 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 2.4582 & -.0241 & -.4377 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0411 & .0411 & .3206 & .3206 & .0000 \\ -.3163 & -.3163 & -2.5974 & -2.5974 & .0000 \\ .1014 & .1014 & -.0699 & -.0699 & .0000 \\ .0003 & -.0003 & -.0004 & .0004 & .0006 \\ .0762 & -.0762 & .5339 & -.5339 & .1144 \\ .0486 & -.0486 & .0071 & -.0071 & -.0544 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \tag{2.5}$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \tag{2.6}$$

$$\begin{bmatrix} v \\ \beta \\ \theta \\ \phi \\ r \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & .0349 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ u \\ w \\ q \\ \beta \\ p \\ r \end{bmatrix} \tag{2.7}$$

The aircraft dynamics described by Equation 2.1 and Equation 2.2 are implemented in the simulation as illustrated in Figure 2.2.

2.3.2 Autoregressive Moving Average Model The CRCA plant and output equations of Equation 2.1 and Equation 2.2 can be represented by an Autoregressive Moving Average (ARMA) model by discretizing the equations for a given sampling period T. The ARMA representation in this thesis is based upon the techniques of Bokor and Keviczky, illustrated in Appendix B, and result in a reduced order difference equation [5, 861-873]. Lijung and Söderström [14, 99-101] show the input-output relationships of Equation 2.1 and Equation 2.2 expressed in terms of a Nth order autoregressive difference equation of the form

$$\begin{aligned} y(kT) &= B_1 u[(k-1)T] - A_1 y[(k-1)T] + \cdots + B_m u[(k-m)T] - \\ &\quad A_n y[(k-n)T] \end{aligned} \tag{2.8}$$

where,

A_1, A_2, B_1 and B_2 are the coefficients or parameters of the system, m is equal to the number of inputs in the system, and n is the order of the difference equation.

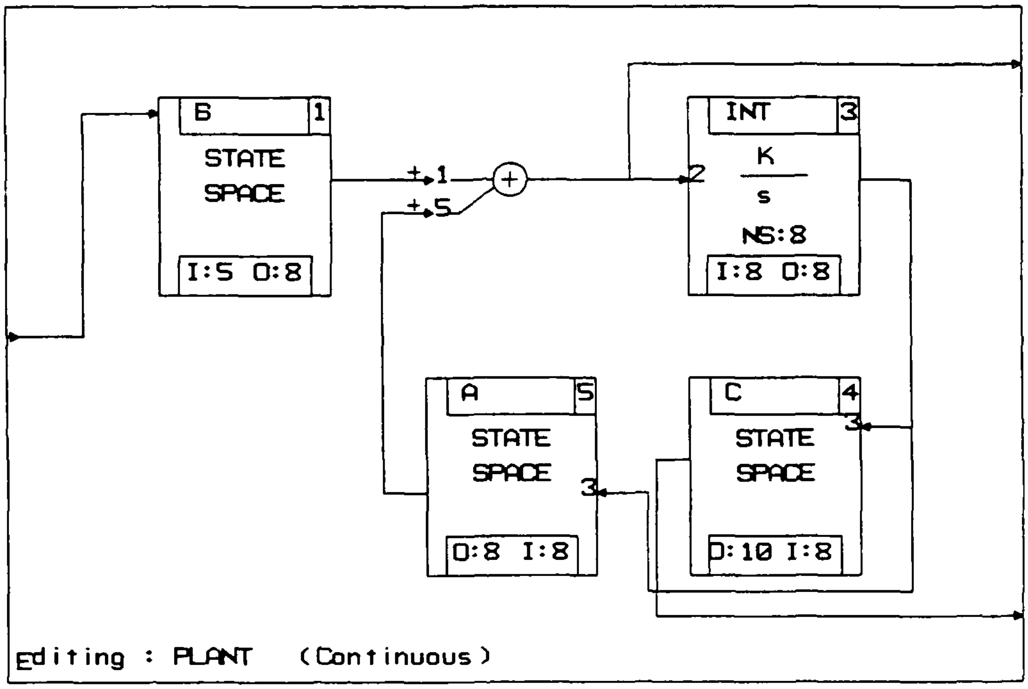


Figure 2.2. Aircraft Plant Representation

Often it is convenient to express Equation 2.8 as

$$y(kT) = \theta^T \phi \tag{2.9}$$

where,

$$\begin{aligned}
\theta^T &= [A_1 \dots A_n \; B_1 \dots B_n] \in R^l \\
\phi^T &= [-y^T[(k-1)T] \; \dots - y^T[(k-n)T] \\
&\quad u^T[(k-1)T] \; \dots u^T[(k-n)T]]
\end{aligned}$$

The vector θ^T is of dimension l which is equal to the number of parameters in the difference equation and ϕ^T is a matrix of the past values of the inputs and outputs of the system. For the CRCA aircraft model of Equation 2.1 and Equation 2.2 the second order ARMA representation consists of the parameter vector θ containing 100 elements and is unique for each flight condition. Table 2.5 contains the ARMA representation for the nominal plant, ACM Entry, and Appendix B contains the ARMA representations for the remaining flight conditions.

Table 2.5. ARMA Representation - A and B Matrices for ACM Entry

$$A_1 = \begin{bmatrix} -1.9766 & .0000 & 2.0200 & .0000 & .0000 \\ .0000 & -1.9877 & .0000 & .0000 & .0000 \\ .0031 & .0000 & -1.9857 & .0000 & .0000 \\ .0000 & -27.8256 & .0000 & -.9750 & -.7002 \\ .0000 & .5755 & .0000 & -.0006 & -.9724 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} .9767 & .0000 & -2.0015 & .0000 & .0000 \\ .0000 & .9898 & .0000 & .0000 & .0000 \\ -.0031 & .0000 & .9881 & .0000 & .0000 \\ .0000 & 27.7994 & .0000 & .0000 & .0000 \\ .0000 & -.6359 & .0000 & .0000 & .0000 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} .0007 & .0007 & .0058 & .0058 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0002 & -.0002 & .0000 \\ .0012 & -.0012 & .0002 & -.0002 & -.0014 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} -.0008 & -.0008 & -.0055 & -.0055 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ -.0006 & .0006 & .0004 & -.0004 & .0001 \\ .0000 & .0000 & .0000 & .0000 & .0000 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table 2.6. Control Surface Position and Rate Limits

Surface	Position Limit (deg)		Rate Limit (deg/sec)
Left Canard	+60	-30	± 100
Right Canard	+60	-30	± 100
Left Trailing Edge	+30	-30	± 100
Right Trailing Edge	+30	-30	± 100
Rudder	+20	-20	± 100

2.4 Actuator Model

In order for simulation results to be valid, actuator dynamics must be included in the design of aircraft control systems. Rate and position limits are often encountered, especially during reconfiguration of failed control surfaces. The control surface actuator model for the canards, trailing edge flaperons, and rudder is approximated by a first order transfer function of the form

$$\delta_{control}(s) = \frac{20}{s+20}\delta_{cmd} \tag{2.10}$$

The PI and PID controller designs in this thesis include incorporation of the first order actuator models. Control surface position and rate limits are listed in Table 2.6 and reflect values for a control surfaces at neutral position.

Each of the linearized equations and state space representations at each flight condition represent control surfaces which are deflected at some value. In the analysis of the aircraft responses, each control surface position is adjusted accordingly to reflect the contribution of initial control surface values obtained from the linearization process. A complete list of the initial control surface positions for each flight condition are contained in Appendix A. Figure 2.3 illustrates the actuator representation of Equation 2.10 in state space form.

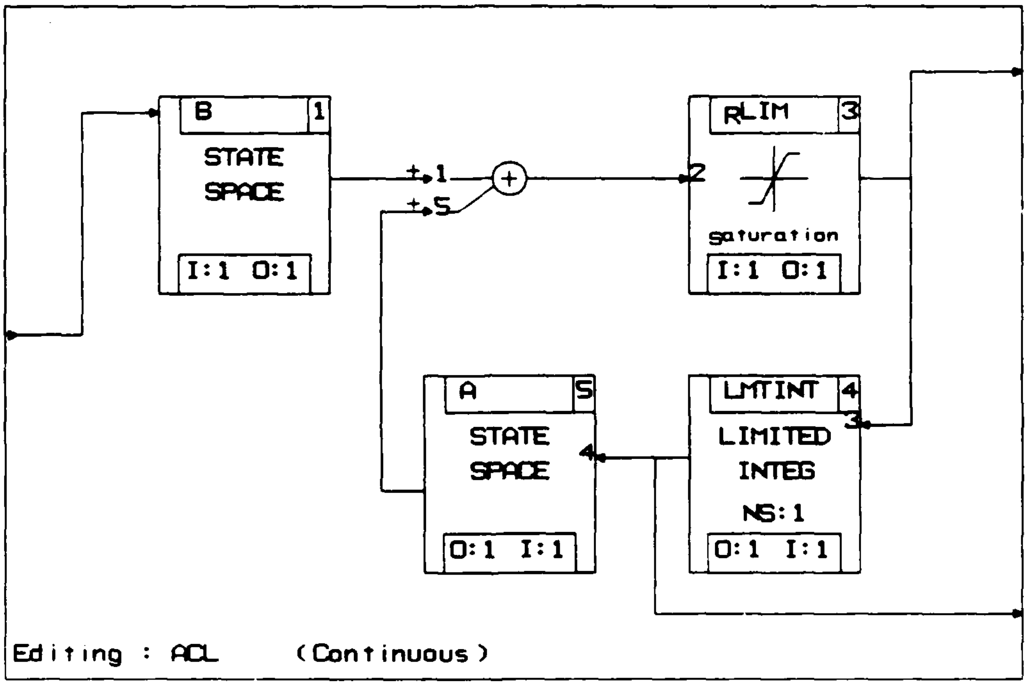


Figure 2.3. Actuator System Representation

The purpose of the limited integrator in the diagram is to stop integration of the actuator deflection rate signal should surface position limits be encountered. This condition prevents the phenomenon known as "integrator wind-up" when the integrator continually tries to calculate control surface position based on the surface deflection rate even though control surfaces are at maximum deflection [27, 602]. The side affects of integrator wind-up include sluggish and inaccurate system response when surface position limits eventually return to within normal operating range.

2.5 Normal Acceleration

The value of the normal acceleration at the pilot station is an important characteristic in evaluating the performance of a maneuver. Aircraft structural limits may be exceeded or the pilot may not be able to physically endure the increase in *g's*. Therefore, all aircraft responses presented in this thesis include monitoring this value. The complete set of mathematical equations needed to calculate the acceleration, A_z, of the center of gravity (C.G.) is contained in reference [3]. However, the normal acceleration, N_z, felt by the pilot must also consider the distance the pilot station is located from the center of gravity. For the CRCA,

$$N_{z_{\text{(Pilot Station)}}} = \frac{-A_{z_{\text{c.g.}}} + (28 \text{ ft})\ddot{\theta}}{g} \text{ (units} = g) \tag{2.11}$$

where,

$g = 32.174$ ft$/sec^2$

$\ddot{\theta}$ = Pitch Acceleration

A_z = Acceleration of C.G.

$$A_{z_{c.g.}} = \dot{w} - (U_0)q + [g\sin(\theta_0)]\theta \tag{2.12}$$

$$\ddot{\theta} = (MSU)u + (MW)w + (MQ)\dot{\theta} + (MWDOT)\dot{w} + (MDCL)\delta_{cl} + (MDCR)\delta_{cr} + (MDTEL)\delta_{tel} + (MDTER)\delta_{ter} \tag{2.13}$$

The coefficients for each flight condition are shown in Appendix F.

2.6 Summary

The Control Reconfigurable Combat Aircraft represents future aircraft design with demonstration of mission and control power developments in canard controlled aircraft. It is described by a set of linear differential equations developed from the equations of motion and stability coefficients. The aircraft models used in this research are linearized about 3 flight conditions, TF/TA, ACM Entry, and ACM Exit and include the effects of actuator dynamics. In preparation for the adaptive control analysis, the autoregressive moving average (ARMA) equivalent representation of aircraft plant dynamics is presented. With the aircraft models defined and the input/output relationships established, the PI controller implementation methodology is described in Chapter 3.

III. Control Law Design

3.1 Introduction

This chapter describes the control algorithm that is implemented in the Proportional plus Integral (PI) controller and parameter adaptive control system shown in Figure 1.1. An alternative design technique, using the Proportional plus Integral plus Derivative (PID) control theory, is developed in Appendix D and provides supplemental information when certain plant characteristics are present. The primary emphasis is placed on the multivariable control law design techniques developed by Professor Brian Porter of the University of Salford, England. These design methods incorporate high gain error-actuated controllers using output feedback. Output feedback is preferred because state variable reconstruction may be difficult, impractical, and accessible plant dynamics are not realizable. Modelling of plants using the input/output data to capture the entire dynamics of the system is essential when designing an adaptive controller. The structure and characteristics of the plant can be modelled by use of the impulse and step response matrix calculations, thus eliminating the need to know the internal mathematical equations that compose the system. The theory presented in this chapter is primarily obtained from references [8] and [17].

3.2 Porter's PI Control Law

The Porter design technique requires describing the plant to be controlled as a set of input and output equations expressed in the state-space form of Equation 2.1 and Equation 2.2

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

where,

A = the continuous time plant matrix (n x n)

B = the continuous time control matrix (n x m) with the rank of B = m

C = the continuous time output matrix (p x n)

x = the state variable vector with n states

u = the control input vector with m inputs

y = the output vector with p outputs

Equation 2.1 and Equation 2.2 may be transformed into a form

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} u(t) \tag{3.1}$$

$$y(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{3.2}$$

where,

$$\begin{aligned} x_2 &= m \text{ x } 1 \\ B_2 &= m \text{ x } m \text{ and has rank } m \\ C_2 &= m \text{ x } m \text{ and has rank } m \end{aligned}$$

The design method used for the controller in this research project requires the number of controlled outputs *y(t)* be equal to the number of control inputs *u(t)*, that is, $m = p$. Furthermore, the design of the system is dependent on the rank of the first Markov parameter, the matrix product CB. When CB has full rank "m", the plant is defined as *regular* and the PI controller is implemented as illustrated in Figure 3.1.

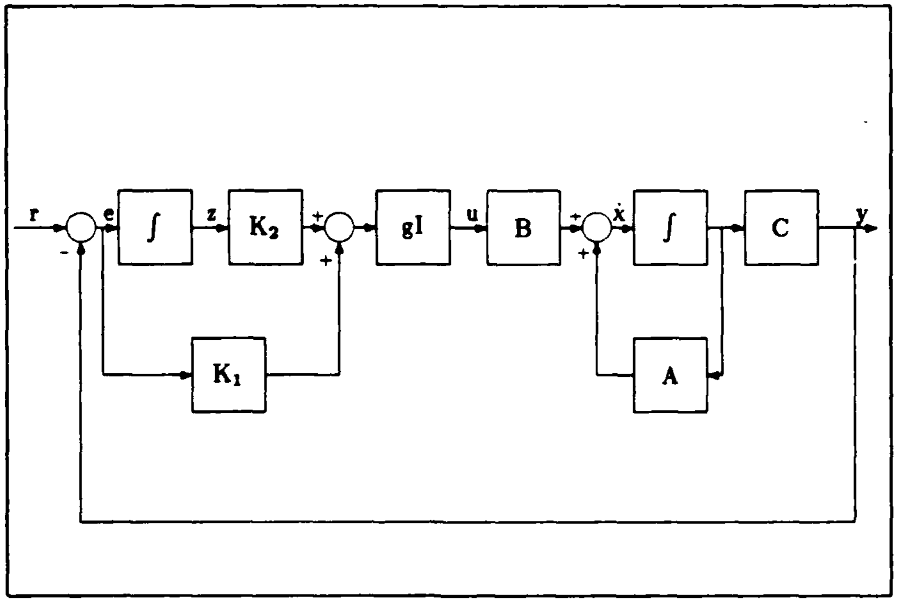


Figure 3.1. PI Controller Regular Plant Design

When CB is rank deficient, the plant is defined as *irregular*. Irregular plant controller design requires either the augmentation of an inner-loop which provides extra measurements for control purposes, Figure 3.2, or a Proportional plus Integral plus Derivative (PID) control law implementation, Appendix D.

The configuration of the system in this thesis provides for an irregular design and incorporates the following design parameters contained in Equation 3.1 and Equation 3.2.

$$x_1 = \begin{bmatrix} \theta & \phi \end{bmatrix}^T \quad (3.3)$$

$$x_2 = \begin{bmatrix} u & w & q & \beta & p & r \end{bmatrix}^T \quad (3.4)$$

$$u(t) = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \quad (3.5)$$

where,

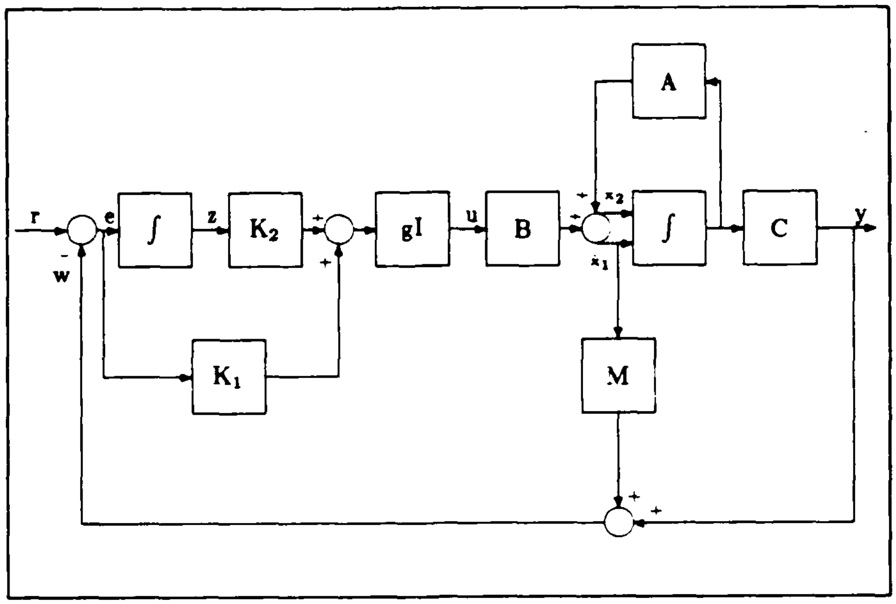


Figure 3.2. PI Controller Irregular Plant Design

δ_{cl} = left canard

δ_{cr} = right canard

δ_{tel} = left trailing edge flaperon

δ_{ter} = right trailing edge flaperon

δ_{rud} = rudder

The output matrix C, composed of linear combinations of the state variables, provides the following five outputs.

v = forward velocity

β = side slip angle

θ = pitch angle

ϕ = bank angle

r = yaw rate

The high-gain controller implementing the proportional plus integral (PI) control law, Figure 3.2, is expressed in the continuous time domain by the relationship

$$u(t) = g[K_1 e(t) + K_2 z(t)] \tag{3.6}$$

where,

$$\begin{aligned} g &= \text{scalar gain} \\ K_1 &= \text{proportional gain matrix (m x m)} \\ K_2 &= \text{integral gain matrix (m x m)} \\ e(t) &= \text{error vector between the input r(t) and output y(t)} \\ z(t) &= \int_0^t e(t)\,\mathrm{dt} \end{aligned}$$

For the discrete PI controller, Equation 3.6 is expressed as

$$u(kT) = (1/T)[K_1 e(kT) + K_2 z(kT)] \tag{3.7}$$

where,

$$\begin{aligned} T &= \text{sampling period (1/f)} \\ K_1 &= \text{proportional gain matrix (m x m)} \\ K_2 &= \text{integral gain matrix (m x m)} \\ e(kT) &= \text{error vector between the r(kT) and y(kT)} \\ z(kT) &= \text{digital integral of the error e(kT)} \\ & \text{approximated by the difference equation} \\ & z[(k+1)T] = z(kT) + (1/T)e(kT) \end{aligned}$$

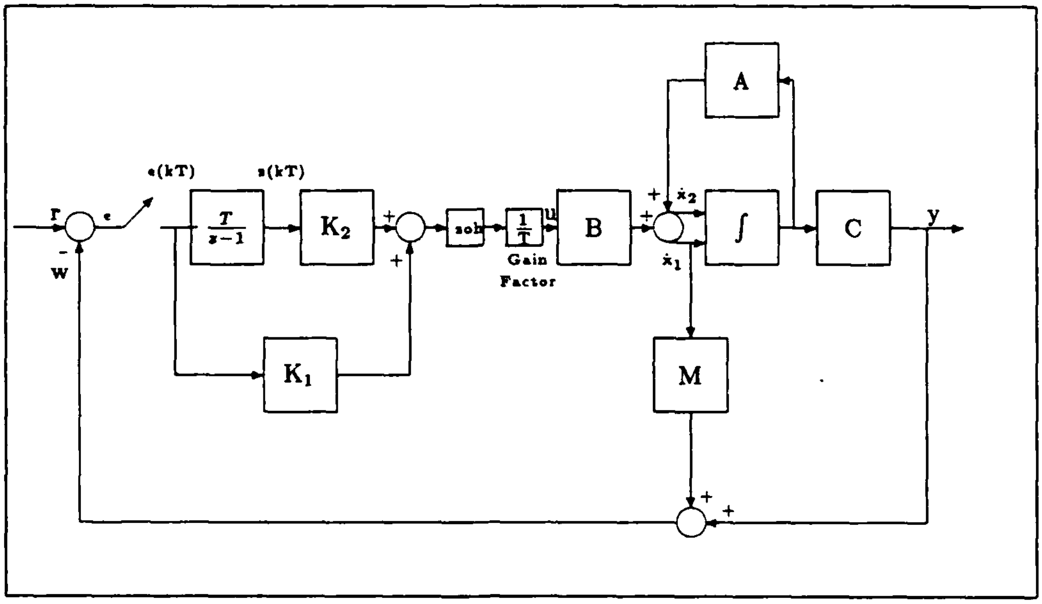


Figure 3.3. Discrete PI Controller - Irregular Plant

[9]

Figure 3.3 illustrates the implementation of Equation 3.7 with the appropriate gain factor and zero order hold (zoh).

3.2.1 Fixed Gain Matrices The form of the B matrix in Equation 3.1 and an investigation of the control law representation of Equation 3.6 as g becomes large, or asymptotically approaches infinity, $g \rightarrow \infty$, develops the performance of the closed-loop system under high gain operation.

In Figure 3.2, the minor loop feedback described by

$$w(t) = y(t) + M\dot{x}_1 \tag{3.8}$$

enables the irregular system to be controllable. Inserting the values obtained from Equation 3.1 and Equation 3.2 into Equation 3.8 yields a new output equation

$$w(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + M \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.9}$$

$$= \begin{bmatrix} (C_1 + MA_{11}) & (C_2 + MA_{12}) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.10}$$

$$= \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.11}$$

For the closed loop tracking system of Figure 3.2, Equations 3.1 through 3.6 and Equation 3.8, are combined to yield the form

$$\begin{bmatrix} \dot{z}(t) \\ \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -F_1 & -F_2 \\ 0 & A_{11} & A_{12} \\ gB_2K_2 & A_{21} - gB_2K_1 & A_{22} - gB_2K_1F_2 \end{bmatrix} \begin{bmatrix} z(t) \\ x_1(t) \\ x_2(t) \end{bmatrix} +$$

$$\begin{bmatrix} I_p \\ 0 \\ gB_2K_1 \end{bmatrix} r(t) \tag{3.12}$$

where,

$$r(t) = \text{vector of command inputs, } (m \text{ x } 1)$$

and,

$$y(t) = \begin{bmatrix} 0 & C_1 & C_2 \end{bmatrix} \begin{bmatrix} z(t) \\ x_1(t) \\ x_2(t) \end{bmatrix} \tag{3.13}$$

The closed-loop transfer function for Equation 3.12 and Equation 3.13 is

$$G(\lambda) = \begin{bmatrix} 0 & C_1 & C_2 \end{bmatrix} \begin{bmatrix} \lambda I_p & F_1 & F_2 \\ 0 & \lambda I_{n-p} - A_{11} & -A_{12} \\ -gB_2K_2 & -A_{21} + gB_2K_1 & \lambda I_p - A_{22} + gB_2K_1F_2 \end{bmatrix}^{-1} \bullet$$

$$\begin{bmatrix} I_p \\ 0 \\ gB_2K_1 \end{bmatrix} \tag{3.14}$$

Equation 3.12 and Equation 3.13 are transformed into block diagonal form

$$\begin{bmatrix} x_s(t) \\ x_f(t) \end{bmatrix} = \begin{bmatrix} A_s & 0 \\ 0 & A_f \end{bmatrix} \begin{bmatrix} x_s(t) \\ x_f(t) \end{bmatrix} + \begin{bmatrix} B_s \\ B_f \end{bmatrix} r(t) \tag{3.15}$$

$$y(t) = \begin{bmatrix} C_s & C_f \end{bmatrix} \begin{bmatrix} x_s(t) \\ x_f(t) \end{bmatrix} \tag{3.16}$$

where the subscript "s" designates association with the slow modes of the system and the subscript "f" corresponds to the fast modes of the system. As $g \to \infty$ the components of Equation 3.15 and Equation 3.16 yield

$$A_s = \begin{bmatrix} -K_1^{-1}K_2 & 0 \\ A_{12}F_2^{-1}K_1^{-1}K_2 & A_{11} - A_{12}F_2^{-1}F_1 \end{bmatrix} \tag{3.17}$$

$$B_s = \begin{bmatrix} 0 \\ A_{12}F_2^{-1} \end{bmatrix} \tag{3.18}$$

$$C_s = \begin{bmatrix} C_2F_2^{-1}K_1^{-1}K_2 & C_1 - C_2F_2^{-1}F_1 \end{bmatrix} \tag{3.19}$$

$$A_f = -gB_2K_1F_2 \tag{3.20}$$

$$B_f = gB_2K_1 \tag{3.21}$$

$$C_f = C_2 \tag{3.22}$$

The measurement matrix M in Equations 3.8 through 3.11 is selected with as few non-zero elements as possible, containing only enough elements so that F_2 has rank m. Reference [8] gives an approach to selecting a measurement matrix for optimum decoupling.

The slow transfer function, determined from Equation 3.17 through Equation 3.19, is

$$\Gamma_s(\lambda) = [C_1 - C_2F_2^{-1}F_1][\lambda I_p - A_{11} + A_{12}F_2^{-1}F_1]^{-1}A_{12}F_2^{-1} \tag{3.23}$$

and contains only the transmission zeros.

The fast transfer function, determined from Equation 3.20 through Equation 3.22, is

$$\Gamma_f(\lambda) = C_2F_2^{-1}[\lambda I_{n-p} + gF_2B_2K_1]^{-1}gF_2B_2K_1 \tag{3.24}$$

The proportional matrix, K_1, from Equation 3.6, is selected to make the fast transfer function, Equation 3.24 diagonal. This is accomplished by selecting a diagonal matrix Σ such that

$$F_2B_2K_1 = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \vdots & \\ \vdots & & \ddots & 0 \\ 0 & & & \sigma_m \end{bmatrix} = \Sigma \tag{3.25}$$

$$K_1 = (F_2B_2)^{-1}\Sigma \tag{3.26}$$

The integral gain matrix, K_2, is assigned using

$$K_2 = \lambda K_1 \tag{3.27}$$

where,

λ = proportional to integral proportionality constant (adjusted to give good transient response)

3.2.2 Step-Response Matrices The PI controller gains K_1 and K_2 previously developed are realizable only when the mathematical model of the plant is expressed in state space form. Should these matrices not be obtainable, the PI controller can be designed using the step response matrix, H(T) and H(∞) [16,17,19,22,24]. This procedure involves applying a unit step input to the open-loop plant and measuring the output response at the sampling time T and as T $\rightarrow$ ∞. Should the open-loop plant be unstable, that is, eigenvalues of the plant exist in the right half plane, this technique of obtaining H(T) and H(∞) is not possible. References [5] and [18] develop a procedure to obtain H(T) and H(∞) from the autoregressive difference equation. The significance of using H(T) is that it is readily obtained from the input-output measurements, thus reflecting the current status of the plant, and serves as a basis for a parameter adaptive system [16, 3]. These matrices are defined as

$$H(T) = \int_0^T Cexp(At)B \, \mathrm{dt} \tag{3.28}$$

$$H(T) = G(0) \approx -CA^{-1}B \tag{3.29}$$

For small sampling periods H(T) $\approx$ T[CB] and the control law of Equation 3.8 becomes

$$u(kT) = \bar{K}_1 e(kT) + \bar{K}_2 z(kT) \tag{3.30}$$

where,

$$\bar{K}_1 = H(T)^T[H(T)H(T)^T]^{-1}\Sigma \tag{3.31}$$

$$\bar{K}_2 = G(0)^T[G(0)G(0)^T]^{-1}\Pi \tag{3.32}$$

$$\Sigma = \text{diagonal weighting matrix } [\sigma_1, \ldots \sigma_m] \tag{3.33}$$

$$\Pi = \text{diagonal weighting matrix } [\pi_1, \ldots \pi_m] \tag{3.34}$$

In the case of plant parameter changes, due to control surface failures, it is essential to provide updated step-response matrices to incorporate into the control law of Equation 3.30.

The behavior of the plant can be modelled at each sampling period T by means of an autoregressive difference equation, Equation 2.8, of the form

$$\begin{aligned} y(kT) &= B_1u[(k-1)T] - A_1y[(k-1)T] + \cdots + B_mu[(k-m)T] - \\ &\quad A_ny[(k-n)T] \end{aligned}$$

where,

A_1, A_2, B_1 and B_2 are the coefficients or parameters of the system, m is equal to the number of inputs in the system, and n is the order of the difference equation. Often it is convenient to express Equation 2.8 as Equation 2.9

$$y(kT) = \theta^T\phi$$

where,

$$\begin{aligned} \theta^T &= [A_1 \ldots A_n \; B_1 \ldots B_n] \in R^l \\ \phi^T &= [-y^T[(k-1)T] \ldots - y^T[(k-n)T] \quad u^T[(k-1)T \ldots u^T[(k-m)T] \end{aligned}$$

The vector θ^T is of dimension l which is equal to the number of parameters in the difference equation and ϕ^T is a matrix of the past values of the inputs and outputs of the system.

By applying the definition of the step-response matrix of Equation 3.28 and analyzing Equation 2.8 at time "T"

$$H(T) = B_1 \tag{3.35}$$

$$G(0) = (I + A_1 + \cdots + A_n)^{-1}(B_1 + B_2 + \cdots + B_m) \tag{3.36}$$

The autoregressive difference coefficients, plant parameters, used in the control law calculations are estimated using a recursive least squares method as described in the next section.

3.3 Parameter Adaptive Algorithm

There are many well know parameter identification algorithms to help solve the problem of identifying plant coefficients needed in the ARMA representation [12,13,14]. Included are maximum likelihood and stochastic approximation, but the least squares method selected in this thesis is a classical technique used by many scientific disciplines. It is conceptually simple and applicable in a wide range of situations and exhibits statistical properties that are as good as those of maximum liklihood method for most practical situations [12, 4]. Parameter estimates are recursively calculated every sample period so new input-output data is used to correct and update the existing estimates. Recall that from Equation 2.9, Section 2.3.2, the representation of the plant difference output equation are put in the form

$$y(kT) = \theta^T \phi$$

where,

$$\begin{aligned}\theta^T &= [A_1 \dots A_n\ B_1 \dots B_m] \in R^l \\ \phi^T &= [-y^T[(k-1)T] \dots - y^T[(k-n)T] \quad u^T[(k-1)T \dots u^T[(k-m)T]\end{aligned}$$

The parameter vector θ, Equation 2.9, is desired to be estimated from the input and output time history. The least squares method chooses an estimate by minimizing the "equation error" term $v(t)$ in the equation

$$y(kT) = \theta^T \phi + v(t) \tag{3.37}$$

By choosing a function of the form

$$V_n(\theta) = 1/N \sum_0^N \alpha_t [y(t) - \theta^T \phi(t)]^2 \tag{3.38}$$

and minimizing with respect to θ gives a least squares estimate of the parameter vector θ. The inclusion of the coefficient α_t allows different weights to be applied to different input-output data. Most often α_t is chosen equal to 1 [14, 17]. If Equation 3.38 is expanded, the estimation of θ can be carried out on-line recursively using the equations

$$\theta(k+1) = \theta(k) + P(k)x(k+1)\gamma(k+1)[y(k+1) - x^T(k+1)\theta(k)] \tag{3.39}$$

$$P(k+1) = P(k) - P(k)x(k+1)\gamma(k+1)x^T(k+1)P(k) \tag{3.40}$$

$$\gamma(k+1) = \frac{1}{[\alpha_t + x^T(k+1)P(k)x(k+1)]} \tag{3.41}$$

$$x(k+1) = [-y(k), \ldots, -y(k-n), u(k), \ldots, u(k-n)] \tag{3.42}$$

where,

$$\begin{aligned} \theta(k) &= \text{plant parameter vector} \\ P(k) &= \text{covariance matrix associated with estimation} \\ x(k+1) &= \text{time history of input/output data} \\ \alpha_t &= \text{weighting parameter} \end{aligned}$$

Often it is desired to consider discounting old information when the process is not constant. Inclusion of a forgetting factor helps the algorithm stay more alert to changes in plant parameters. The forgetting factor γ modifies Equation 3.40 and Equation 3.41 to yield

$$P(k+1) = \frac{1}{\gamma}[P(k) - P(k)x(k+1)\gamma(k+1)x^T(k+1)P(k)] \tag{3.43}$$

$$\gamma(k+1) = \frac{1}{[\gamma\alpha_t + x^T(k+1)P(k)x(k+1)]} \tag{3.44}$$

To illustrate the dimensions and characteristics of Equation 3.39 through Equation 3.42, consider the following system with 2 inputs, 2 outputs and described by a 2nd order linear difference equation.

$$\begin{aligned} Y(k) &= \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} Y(k-1) - \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} Y(k-2) + \\ & \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} U(k-1) + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} U(k-2) \end{aligned} \tag{3.45}$$

This is expanded to the form,

$$\phi(k) = \begin{bmatrix} -y_1(k-1) & 0 \\ -y_2(k-1) & 0 \\ 0 & -y_1(k-1) \\ 0 & -y_2(k-1) \\ -y_1(k-1) & 0 \\ -y_2(k-1) & 0 \\ 0 & -y_1(k-1) \\ 0 & -y_2(k-1) \\ u_1(k-1) & 0 \\ u_2(k-1) & 0 \\ 0 & u_1(k-1) \\ 0 & u_2(k-1) \\ u_1(k-1) & 0 \\ u_2(k-1) & 0 \\ 0 & u_1(k-1) \\ 0 & u_2(k-1) \end{bmatrix} \quad \theta = \begin{bmatrix} A_{11}^{Y(k-1)} \\ A_{12}^{Y(k-1)} \\ A_{21}^{Y(k-1)} \\ A_{22}^{Y(k-1)} \\ A_{11}^{Y(k-2)} \\ A_{12}^{Y(k-2)} \\ A_{21}^{Y(k-2)} \\ A_{22}^{Y(k-2)} \\ B_{11}^{U(k-1)} \\ B_{12}^{U(k-1)} \\ B_{21}^{U(k-1)} \\ B_{22}^{U(k-1)} \\ B_{11}^{U(k-2)} \\ B_{12}^{U(k-2)} \\ B_{21}^{U(k-2)} \\ A_{22}^{U(k-2)} \end{bmatrix} \tag{3.46}$$

3.4 Summary

This chapter presents the development of the Proportional Plus Integral, PI, control law developed by Professor Brian Porter. Procedures for the selection of the controller gains K_1 and K_2 are outlined based upon the available plant models, either known A,B and C matrices or an ARMA model representation using the step-response matrices. The chapter concludes with a summary of the recursive least squares parameter estimation algorithm, and an illustration of the format needed for proper implementation. The design process, incorporating the mathematical developments presented in this chapter, for the nominal flight condition is presented in Chapter 4.

IV. Design Procedure

4.1 Introduction

Chapter 3 provides a basis for understanding the mathematical relationships needed to validate the controller and system response. This chapter describes the specific design procedure needed to implement the fixed gain Proportional plus Integral (PI) controller for the CRCA. The nominal plant, ACM Entry, serves as a basis for the controller design with a summary of design parameters tabulated for the remaining flight conditions in Chapter 5. Since a number of user defined controller design parameters are selected for a desired output, many trial and error simulations are required. A powerful simulation tool called $\mathrm{MATRIX_x}$ is invaluable in minimizing the parameter selection process [15].

4.2 Plant Considerations

Before designing the PI controller using the Porter method, several mathematical relationships concerning the plant must be checked. This is particularly important if the plant A and B matrix have been modified using simplifying assumptions, such as those used in this thesis for reducing the number of control surfaces. Of prime importance is the controllability and observability of plant state variables. Secondly, to insure overall stability, system transmission zeros must lie in the open half left s-plane, and not at the origin. Also, the open-loop stability of the plant is checked from the eigenvalues of the A matrix. Finally, the control input needed to perform a desired maneuver must be checked because the control surfaces may not be able to deflect sufficiently to meet the input demands. Checking this relationship is *essential* prior to designing the controller.

4.2.1 Controllability and Observability The eight states associated with the aircraft plant matrices of Table 2.4 must be completely *controllable* and *observable*.

Controllability and observability for this design is made using functions resident in MATRIX_x [8,15]. To insure the input $u(t)$ has an effect on each of the eight states of the plant, the *controllability* matrix M_c is formed from the state-space matrices and must have rank n, where n equals the number of state variables. The *controllability* matrix is formed using the relationship,

$$\text{Rank } M_c = \text{Rank } [B \mid AB \mid \ldots \mid A^{n-m}B] = n \tag{4.1}$$

Likewise, the *observability* matrix M_o must have rank equal to the number of state variables in the system and have the property such that

$$\text{Rank } M_o = \text{Rank } [C^T \mid A^T C^T \mid \ldots \mid (A^T)^{n-m} C^T] = n \tag{4.2}$$

For this design the plant *is* both *observable* and *controllable.*

4.2.2 Transmission Zeros and Open-Loop Eigenvalues Careful consideration must be given to the choice of the outputs used in feedback as the location of the system transmission zeros, if any, is highly dependent on this selection. Transmission zero location is important because certain closed-loop poles of the plant will migrate to these locations as the gain of the system is increased. Closed-loop stability is enhanced by consideration of these locations. By definition, the transmission zeros of the system are defined in the s-plane for a controllable and observable system as the eigenvalues associated with the determinant of the system matrix when the number of inputs m is equal to the number of outputs l [23],

$$Z(s) = \begin{vmatrix} sI - A & -B \\ C & D \end{vmatrix} \tag{4.3}$$

and must reside in the left half s-plane. This is accomplished by selecting combinations of the output, or elements of the C matrix, that yield the desired eigenvalues.

Transmission zeros at the origin are usually present when rates or accelerations are used in feedback. Although PI control has been successfully implemented with systems containing transmission zeros located at the origin [2,16], it is highly desirable to avoid these locations so complete functional controllability of the overall plant is obtained. The idea of functional controllability can be illustrated if the output relationship of the plant in the s-plane is described by the equation

$$Y(s) = G(s)U(s) \tag{4.4}$$

and the final value theorem,

$$\lim_{s \to 0} sF(s) = \lim_{t \to \infty} f(t) \tag{4.5}$$

is applied to Equation 4.4 for a step input *1/s*. Thus at steady state

$$Y_{ss} = G(0)U_{ss} \tag{4.6}$$

If the rank of the determinant of $G(0)$ is equal to m, where m is the number of control inputs, then *functional controllability* is assured because any combinations of the input will fall into the "range" space of $U(s)$. If the determinant of $G(0) =$ 0 (rank $< m$) then the plant is functionally *uncontrollable.* This relationship can also be expressed in terms of Equation 4.3 evaluated as $s \rightarrow 0$ where,

$$Z(0) = 0 \tag{4.7}$$

$$\textit{iff det } -CA^{-1}B = 0 \tag{4.8}$$

$$\textit{iff det } G(0) = 0 \tag{4.9}$$

For this design, the output matrix C is selected as

$$\begin{bmatrix} V \\ \beta \\ \theta \\ \phi \\ r \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0.0349 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ u \\ w \\ q \\ \beta \\ p \\ r \end{bmatrix} \tag{4.10}$$

and results in one transmission zero at -1.6844.

The open-loop eigenvalues of the system reflect the stability of the modes of the system, i.e. dutch roll and spiral divergence. If the plant is open-loop unstable or non-minimum phase, often special compensation is necessary before a controller is implemented. The measurement matrix M used in the *irregular* PI controller design provides the feedback necessary to stabilize the system. For the CRCA and associated flight conditions the open-loop eigenvalues are listed in Table 4.1 and indicate open-loop instability in all cases.

4.2.3 Transfer Functions The s-plane analysis of the open-loop system can give tremendous insight into the characteristics of the plant. Not only can the system transmission zeros be determined, but also determination of the steady-state relationships between the desired output and the control input needed to sustain the chosen response. Equation 4.4 describes the output relationship of the plant in the s-plane as

$$Y(s) = G(s)U(s)$$

Equation 4.4 is rearranged into the form

Table 4.1. Open-Loop Eigenvalues*

ACMENTRY		ACM30TL	
$-3.3281D+00$	$+1.7781D-17i$	$-3.3675D+00$	$-1.9104D-16i$
$+1.6669D+00$	$+3.0815D-16i$	$+1.6470D+00$	$-8.9116D-17i$
$-7.7893D-03$	$-4.9532D-02i$	$-2.2664D-03$	$+3.0974D-02i$
$-7.7893D-03$	$+4.9532D-02i$	$-2.2664D-03$	$-3.0974D-02i$
$-2.0156D+00$	$+2.4113D-16i$	$-2.8559D-01$	$+1.9789D+00i$
$-5.6234D-02$	$+9.3644D-17i$	$-2.0535D+00$	$-3.1495D-16i$
$-2.4818D-01$	$+1.8323D+00i$	$-2.8559D-01$	$-1.9789D+00i$
$-2.4818D-01$	$-1.8323D+00i$	$-5.4349D-02$	$-5.9023D-17i$

ACMEXIT		ACM50CL	
$-2.7472D+00$	$+4.9619D-17i$	$-6.5718D-01$	$+1.7084D+00i$
$-1.7162D+00$	$+7.2274D-17i$	$+1.0563D+00$	$+4.8393D-17i$
$-6.5718D-01$	$-1.7084D+00i$	$-4.0576D-03$	$-3.5425D-02i$
$+4.7183D-01$	$-6.1547D-17i$	$-4.0576D-03$	$+3.5425D-02i$
$-3.0588D-01$	$+3.0117D-17i$	$-2.6137D-01$	$+1.8124D+00i$
$-2.1388D-01$	$-4.6500D-18i$	$-2.0320D+00$	$-1.4761D-16i$
$+2.2007D-02$	$+3.6259D-02i$	$-4.9269D-02$	$-2.3771D-17i$
$+2.2007D-02$	$-3.6259D-02i$	$-2.6137D-01$	$-1.8124D+00i$

where,

ACMENTRY = ACM Entry (nominal flight condition)

ACM30TL = 30 percent loss of effectiveness of the left trailing edge - ACM Entry

ACM50CL = 50 percent loss of effectiveness of the left canard - ACM Entry

ACMEXIT = ACM Exit flight condition

TFTA = TF/TA flight condition

TFTA30TL = 30 percent loss of effectiveness of the left trailing edge - TF/TA

TFTA50CL = 50 percent loss of effectiveness of the left canard - TF/TA

* For all subsequent tables, D represents that value calculated with double precision.

Table 4.1 Open-Loop Eigenvalues (cont)

TFTA		TFTA30TL	
$-2.4418D+00$	$+4.4400D+00i$	$-7.0784D+00$	$-9.8476D-17i$
$-2.4418D+00$	$-4.4400D+00i$	$+2.1920D+00$	$+2.1601D-16i$
$+2.3551D-02$	$-2.6317D-15i$	$-1.6299D-02$	$-7.7358D-03i$
$-5.8254D-02$	$+3.1522D-15i$	$-1.6299D-02$	$+7.7358D-03i$
$-7.5589D-01$	$+2.9853D+00i$	$-5.4025D+00$	$+1.8562D-16i$
$-4.5709D-02$	$-4.1054D-16i$	$-7.4601D-01$	$+2.9732D+00i$
$-5.4284D+00$	$-1.1840D-16i$	$-7.4601D-01$	$-2.9732D+00i$
$-7.5589D-01$	$-2.9853D+00i$	$-4.5436D-02$	$+0.0000D+00i$

TFTA50CL	
$-6.7047D+00$	$-1.0750D-18i$
$+9.5544D-01$	$-3.2592D-16i$
$-1.9358D-02$	$+2.6849D-02i$
$-1.9358D-02$	$-2.6849D-02i$
$-6.1783D+00$	$+4.1319D-17i$
$-8.4840D-01$	$+3.0370D+00i$
$-8.4840D-01$	$-3.0370D+00i$
$-5.1854D-02$	$+0.0000D+00i$

where,

ACMENTRY = ACM Entry (nominal flight condition)

ACM30TL = 30 percent loss of effectiveness of the left trailing edge - ACM Entry

ACM50CL = 50 percent loss of effectiveness of the left canard - ACM Entry

ACMEXIT = ACM Exit flight condition

TFTA = TF/TA flight condition

TFTA30TL = 30 percent loss of effectiveness of the left trailing edge - TF TA

TFTA50CL = 50 percent loss of effectiveness of the left canard - TF/TA

$$U(s) = G(s)^{-1}Y(s) \tag{4.11}$$

If the final value theorem of Equation 4.5,

$$\lim_{s \to 0} sF(s) \quad = \quad \lim_{t \to \infty} f(t)$$

is applied to Equation 4.11, then

$$U_{ss} = G(0)^{-1}Y_{ss} \tag{4.12}$$

Equation 4.12 provides the relationship needed to evaluate the steady state control input, U_{ss}, needed to obtain a given steady state value of the output, Y_{ss}. The CRCA plant transfer function matrix $G(s)$ is of the form,

$$G(s) = \begin{bmatrix} \frac{V}{V_{cmd}} & \frac{\beta}{V_{cmd}} & \frac{\theta}{V_{cmd}} & \frac{\phi}{V_{cmd}} & \frac{r}{V_{cmd}} \\ \frac{V}{\beta_{cmd}} & \frac{\beta}{\beta_{cmd}} & \frac{\theta}{\beta_{cmd}} & \frac{\phi}{\beta_{cmd}} & \frac{r}{\beta_{cmd}} \\ \frac{V}{\theta_{cmd}} & \frac{\beta}{\theta_{cmd}} & \frac{\theta}{\theta_{cmd}} & \frac{\phi}{\theta_{cmd}} & \frac{r}{\theta_{cmd}} \\ \frac{V}{\phi_{cmd}} & \frac{\beta}{\phi_{cmd}} & \frac{\theta}{\phi_{cmd}} & \frac{\phi}{\phi_{cmd}} & \frac{r}{\phi_{cmd}} \\ \frac{V}{r_{cmd}} & \frac{\beta}{r_{cmd}} & \frac{\theta}{r_{cmd}} & \frac{\phi}{r_{cmd}} & \frac{r}{r_{cmd}} \end{bmatrix} \tag{4.13}$$

and evaluated as $s \rightarrow 0$.

The steady state transfer function and its inverse for the nominal flight condition are listed in Table 4.2.

Commanded maneuvers for the CRCA aircraft control system analysis are a 45 degree banked coordinated turn and 2 degree second pitch tracking. Blakelock defines a coordinated turn as a command that produces a bank angle (ϕ) and a yaw rate (r) [3]. The yaw rate is determined from the relationship

$$r\left(\frac{deg}{sec}\right) = \frac{g}{V}\sin(\phi)57.3\,\frac{deg}{rad} \tag{4.14}$$

Table 4.2. Steady State Transfer Functions - ACM Entry

$$G(0) = \begin{bmatrix} 2.4333D+2 & 2.4333D+2 & -2.1279D+2 & -2.1279D+2 & 3.9683D-17 \\ 5.6314D-3 & -5.6314D-3 & 2.2037D-2 & -2.2037D-2 & 1.8004D-03 \\ -8.4389D-2 & -8.4389D-2 & 8.6413D-2 & 8.6413D-2 & -1.2683D-13 \\ 3.9791D+0 & -3.9791D+0 & 3.9664D+0 & -3.9664D+0 & -3.1916D+00 \\ 1.4294D-1 & -1.4294D-1 & 1.4026D-1 & -1.4026D-1 & -1.1439D-01 \end{bmatrix}$$

$$G(0)^{-1} = \begin{bmatrix} 1.4075D-02 & 4.2024D+1 & 3.4660D+1 & -1.9137D+1 & 5.3458D+2 \\ 1.4075D-02 & -4.2024D+1 & 3.4660D+1 & 1.9137D+1 & -5.3458D+2 \\ 1.3745D-02 & 6.9626D+0 & 3.9634D+1 & 6.2206D+0 & -1.7345D+2 \\ 1.3745D-02 & -6.9626D+0 & 3.9634D+1 & -6.2206D+0 & 1.7345D+2 \\ 4.7110D-12 & 1.2209D+2 & 1.1951D-8 & -3.2569D+1 & 9.0186D+2 \end{bmatrix}$$

Table 4.3. r Command - 45° Bank Angle

Flight Condition	Velocity ft/sec	Gravity ft/sec^2	Yaw Rate deg/sec
ACM Entry	895.0	32.17	1.457
ACM Exit	263.0	32.17	4.956
TF/TA	1004.9	32.17	1.297

where,

g = the gravitational constant ($32.174 ft/sec^2$)

$\mathbf{V}$ = the forward velocity of the aircraft

ϕ = the desired bank angle (45° = *0.7854 radians)*

Table 4.3 gives the corresponding yaw rate (r) for each of the flight conditions considered in this thesis when the bank angle ϕ is equal to 45 degrees.

For a 45 degree coordinated turn, the output vector Y_{ss} has the form

$$Y_{ss} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.7854 \\ 0.0254 \end{bmatrix} \text{ corresponding to } \begin{bmatrix} V \\ \beta \\ \theta \\ \phi \\ r \end{bmatrix} \tag{4.15}$$

where units are in *rad* and *rad/sec*.

For a 6 degree θ pitch angle (2 deg/sec for 3 seconds) the output vector Y_{ss} has the form

$$Y_{ss} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.1047 \\ 0.0000 \\ 0.0000 \end{bmatrix} \text{ corresponding to } \begin{bmatrix} V \\ \beta \\ \theta \\ \phi \\ r \end{bmatrix} \tag{4.16}$$

where units are in *rad* and *rad/sec*.

Therefore the coordinated turn maneuver results in the steady state control input vector, U_{ss}, having the form

$$U_{ss} = \begin{bmatrix} -1.4517 \\ 1.4517 \\ 0.4801 \\ -0.4801 \\ -2.6726 \end{bmatrix} \text{ corresponding to } \begin{bmatrix} \delta_{cl} \\ \delta_{cr} \\ \delta_{tel} \\ \delta_{ter} \\ \delta_{rud} \end{bmatrix} \tag{4.17}$$

where units of the control surface deflections are in *deg*.

For the 6 degree steady state pitch angle θ command, U_{ss} has the form

$$U_{ss} = \begin{bmatrix} 3.6296 \\ 3.6296 \\ 4.1505 \\ 4.1505 \\ 0.0000 \end{bmatrix} \text{ corresponding to } \begin{bmatrix} \delta_{cl} \\ \delta_{cr} \\ \delta_{tel} \\ \delta_{ter} \\ \delta_{rud} \end{bmatrix} \tag{4.18}$$

where units are in *deg*.

All steady state surface deflections are well within the position limits for the nominal flight condition. The steady state control surface deflections for all the flight conditions, including failure models, for the coordinated turn and pitch tracking commands are listed in Table 4.4 and Table 4.5 respectively.

The control surface deflection limits are exceeded for these commands at ACM Exit. Therefore, the commands must be reduced to stay within the capability of the aircraft. To evaluate the controller design at this flight condition, 2 degrees of β is commanded. This results in a steady state U_{ss} indicated in Table 4.6.

4.3 Continuous Controller Design Variables

The initial selection of the design variables, K_1, K_2, g and M, for the controller, illustrated by Figure 3.2, is accomplished by following the theory developed in Chapter 3. Since the matrix product C_2B_2 does not have full rank (rank deficiency is 2), it is necessary to add additional measurements and include minor loop feedback. The first step is to select the measurement matrix M with as few $non - zero$ elements as possible so the matrix

$$F_2 = [C_2 + MA_{12}] \tag{4.19}$$

has full rank m. Simultaneously the matrix product $C_2F_2^{-1}$ must be made diagonal. This selection will insure optimum decoupling of the outputs and is accomplished as follows [8]:

Table 4.4. Steady State Control Inputs - 45° Banked Turn

Flight Condition	δ_{cl} (deg)	δ_{cr} (deg)	δ_{tel} (deg)	δ_{ter} (deg)	δ_{rud} (deg)
ACMENTRY	-1.4570	1.4570	0.4801	-0.4801	-2.6726
ACM30TL	1.7181	-1.7259	0.0613	-0.0455	-2.8484
ACM50CL	0.8539	1.6300	1.1226	0.7327	-1.2311
ACMEXIT	72.8110	18.2869	82.2557	114.0228	61.1379
TFTA	-0.3664	0.3664	0.2181	-0.2181	-0.4313
TFTA30TL	-0.4190	0.3684	0.3435	-0.2782	-0.8498
TFTA50CL	-0.9488	0.1031	-0.1306	-0.4729	-0.6676

where,

ACMENTRY = ACM Entry (nominal flight condition)

ACM30TL = 30 percent loss of effectiveness of the left trailing edge - ACM Entry

ACM50CL = 50 percent loss of effectiveness of the left canard - ACM Entry

ACMEXIT = ACM Exit flight condition

TFTA = TF/TA flight condition

TFTA30TL = 30 percent loss of effectiveness of the left trailing edge - TF/TA

TFTA50CL = 50 percent loss of effectiveness of the left canard - TF/TA

Table 4.5. Steady State Control Inputs - θ_{cmd}

Flight Condition	δ_{cl} (deg)	δ_{cr} (deg)	δ_{tel} (deg)	δ_{ter} (deg)	δ_{rud} (deg)
ACMENTRY	3.6296	3.6296	4.1505	4.1505	0.0000
ACM30TL	8.0894	8.3053	16.3492	9.5822	0.0000
ACM50CL	-48.9174	-10.9709	-13.9138	-18.0445	-10.9709
ACMEXIT	-49.3560	-43.2737	-95.9565	-85.9518	-9.2166
TFTA	3.1148	3.4404	4.9591	4.7778	-0.2836
TFTA30TL	2.4073	2.3993	6.0560	3.5177	-0.0040
TFTA50CL	6.6173	3.3265	5.4041	5.2893	0.0179

where,

ACMENTRY = ACM Entry (nominal flight condition)

ACM30TL = 30 percent loss of effectiveness of the left trailing edge - ACM Entry

ACM50CL = 50 percent loss of effectiveness of the left canard - ACM Entry

ACMEXIT = ACM Exit flight condition

TFTA = TF/TA flight condition

TFTA30TL = 30 percent loss of effectiveness of the left trailing edge - TF/TA

TFTA50CL = 50 percent loss of effectiveness of the left canard - TF/TA

Table 4.6. Steady State Control Inputs - 2° of β

Flight Condition	δ_{cl} (deg)	δ_{cr} (deg)	δ_{tel} (deg)	δ_{ter} (deg)	δ_{rud} (deg)
ACMEXIT	0.9441	-0.8708	1.6151	-1.4155	1.2347

1. Form the matrix

$$B^* = \begin{bmatrix} c_1^T A_{11}^{d_1} A_{12} \\ \vdots \\ c_m^T A_{11}^{d_m} A_{12} \end{bmatrix} \tag{4.20}$$

where m is the number of control inputs, c_i^T is the ith row of C_1 and

$$d_i = min\left[j : c_i^T A_{11}^j A_{12} \neq 0, j = 0, 1, \ldots, n-1\right] \tag{4.21}$$

Equation 4.21 specifies that d_i is the smallest value of j for which $c_i^T A_{11}^j A_{12} \neq 0$. The permissible values of j are $0, 1, \ldots, n-1$. If all the values of j result in $c_i^T A_{11}^j A_{12} = 0$, then use $d_i = n - 1$, where n = dimension of A_{11}.

2. Form $F_2 = C_2 + MA_{12}$ using a general form of the matrix $M = [m_{ij}]$. The elements m_{ij} appearing in F_2 are permitted nonzero values only if B^* has a non zero element in a corresponding position. All other elements of M are set equal to zero.

For this design, Equation 4.20, is formed using

$$C_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C_2 = \begin{bmatrix} 1 & .0349 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.22}$$

$$A_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad A_{12} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.0349 \end{bmatrix} \tag{4.23}$$

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.24}$$

$$B_2 = \begin{bmatrix} .0411 & .0411 & .3206 & .3206 & .0000 \\ -.3163 & -.3163 & -2.5974 & -2.5974 & .0000 \\ .1014 & .1014 & -.0699 & -.0699 & .0000 \\ .0003 & -.0003 & -.0004 & .0004 & .0006 \\ .0762 & -.0762 & .5339 & -.5339 & .1144 \\ .0486 & -.0486 & .0071 & -.0071 & -.0544 \end{bmatrix} \tag{4.25}$$

Since $A_{11} = 0$, Equation 4.20 yields the values $d_1 = d_2 = d_3 = d_4 = d_5 = 0$. Then B^* is formed

$$B^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.0349 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.26}$$

The matrix C_2 in Equation 4.19 has a dimension 5 x 6. Since the dimension of A_{12} is 2 x 6, the necessary dimension of M is 5 x 2. Using M of the form

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \\ m_{41} & m_{42} \\ m_{51} & m_{52} \end{bmatrix} \tag{4.27}$$

the matrix F_2 of Equation 4.19 is

$$F_2 = \begin{bmatrix} 1 & 0.0349 & m_{11} & 0 & m_{12} & 0.0349m_{12} \\ 0 & 0 & m_{21} & 1 & m_{22} & 0.0349m_{22} \\ 0 & 0 & m_{31} & 0 & m_{32} & 0.0349m_{32} \\ 0 & 0 & m_{41} & 0 & m_{42} & 0.0349m_{42} \\ 0 & 0 & m_{51} & 0 & m_{52} & 1.0349m_{52} \end{bmatrix} \tag{4.28}$$

The assignable elements of F_2 are permitted nonzero values only if B^* has a nonzero element in the same position. Thus, it is required that $m_{11} = m_{12} = m_{21} = m_{22} =$

$m_{32} = m_{41} = m_{51} = m_{52} = 0$. The matrix $C_2F_2^{-1}$ must be diagonal, but since F_2 is dimensioned 5 x 6, the pseudo inverse of the matrix product is used.

$$C_2F_2^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.29}$$

The matrix M elements m_{31} and m_{42} do not appear in Equation 4.29 and must be assigned during the design process. The resulting matrices from Equation 3.11 are

$$F_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} 1 & 0.0349 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & m_{31} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{42} & 0.0349m_{42} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.30}$$

$$F_2B_2 = \begin{bmatrix} .0301 & .0201 & .2299 & .2299 & .0000 \\ .0003 & -.0003 & -.0004 & .0004 & .0000 \\ .1014m_{31} & .1014m_{31} & -.0699m_{31} & -.0699m_{31} & .0000 \\ 1.0017m_{42} & -1.0017m_{42} & -1.0003m_{42} & -1.0003m_{42} & .9981m_{42} \\ .0486 & -.0486 & .0071 & -.0044 & -.0544 \end{bmatrix} \tag{4.31}$$

Rank $F_2B_2 = m$ for positive m_{31} and m_{42}

The slow eigenvalue set, determined from Equation 3.23 is

$$Eigenvalues_{slow} = \left|\lambda I_{n-p} - \left[A_{11} + A_{12}F_2^{-1}F_1\right]\right| = 0 \tag{4.32}$$

and results in two transmission zeros, $S_1 = -1/m_{31}$ and $S_2 = -1/m_{42}$. A stable system is insured for positive values of m_{31} and m_{42}. For the design in this thesis, the value of $m_{31} = 0.4$ and $m_{42} = 0.2$ insured adequate gain and phase margin in

the open-loop system for the θ vs θ_{cmd} and ϕ vs ϕ_{cmd} respectively. As a final check of the location of all the system transmission zeros, Equation 4.3 is formed for the composite system with the F matrix replacing the C:

$$z(s) = \begin{vmatrix} sI - A & -B \\ F & D \end{vmatrix} \tag{4.33}$$

where $F = [F_1 \ F_2]$. For the ACM Entry flight condition, the system transmission zeros now lie at -2.5,-5.0, and -1.6844, confirming the proper selection of the measurement matrix.

Recall that Equation 3.26 and Equation 3.27 give the relationship for K_1 and K_2 as

$$K_1 = (F_2 B_2)^{-1} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_5 \end{bmatrix}$$

and

$$K_2 = \lambda K_1$$

The complete system of Figure 3.2 is simulated using MATRIX_x with initial values of $\Sigma = diag[\sigma_1 \ldots \sigma_5]$ for Equation 3.26, λ for Equation 3.27, g, and nominal initial values for m_{31} and m_{42}. Through a trial and error process, appropriate values of these parameters are selected to yield an acceptable response. However, the selection of the design parameters is not completely arbitrary. Initially, the Σ weighting matrix is chosen as the identity matrix. If the system response is unstable, reducing the gain g is usually sufficient to get a stable response. Once a stable response is obtained, each weighting element of the Σ matrix is adjusted to fine tune the desired tracking output. For instance, σ_1 adjustments primarily affect the first output chosen in the C matrix, or the velocity output. The σ_2 weighting

Table 4.7. Continuous Time System Design Parameters - ACMENTRY

Flight Condition	σ_1	σ_2	σ_3	σ_4	σ_5	λ	g	m_{31}	m_{42}
ACMENTRY	2	0.01	0.3	2	4	0.1	5	0.40	0.20

Table 4.8. Continuous Gain Matrices - ACMENTRY

$$K_1 = \begin{bmatrix} 2.7500 & 5.3158 & 33.9250 & 3.1158 & 26.0293 \\ 2.7500 & -5.3158 & 33.9250 & -3.1158 & -26.0293 \\ 3.9893 & -1.7279 & -4.4350 & 8.0975 & -0.9249 \\ 3.9893 & 1.7279 & -4.4350 & -8.0975 & 0.9249 \\ 0.0000 & 9.0470 & 0.0000 & 7.6808 & -27.2626 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 0.2750 & 0.5316 & 3.3925 & 0.3116 & 2.6029 \\ 0.2750 & -0.5316 & 3.3925 & -0.3116 & -2.6029 \\ 0.3989 & -0.1728 & -0.4435 & 0.8097 & -0.0925 \\ 0.3989 & 0.1728 & -0.4435 & -0.8097 & 0.0925 \\ 0.0000 & 0.9047 & 0.0000 & 0.7681 & -2.7263 \end{bmatrix}$$

element shapes the response of the second output or the β output, and so forth. Increasing the value of σ_i increases the damping of the response, increasing the value of λ tends to reduce the time to reach steady state output values, increasing the measurement matrix values increases the damping for their corresponding output, and increasing g typically speeds up the response and increases control surface deflection and rates. The continuous time design parameters for the ACM Entry flight condition are listed in Table 4.7 and the corresponding gain matrices K_1 and K_2 are listed in Table 4.8.

The final selection of the design parameters includes adjustment to yield an adequate phase and gain margin for the desired system. Although determining gain and phase margins is more difficult for MIMO systems, and often can give misleading results due to coupling in the output responses, useful information is nevertheless obtained from the frequency domain analysis. For statically unstable

aircraft, the military specification MIL-F-9490D, specifies the gain margin to be $\pm$ 6.0 dB and the phase margin to be $\pm$ 45 degrees. For these types of aircraft the system will have a low and high frequency gain margin. At the low frequency crossover frequency of -180 degrees the gain margin should be smaller than -6.0 dB. At the high frequency crossover of -180 degrees the gain margin should be greater than +6.0 dB. Since the CRCA is a statically unstable aircraft, this unique characteristic of the gain and phase margins surfaces in the θ vs θ_{cmd} output in the ACM Entry flight condition. Figure 4.1 through Figure 4.10 represent the Nichols plots for the compensated system with final design parameters. These plots are extremely useful and quickly illustrate the gain and phase margins. Figure 4.5 shows the characteristic described in MIL-F-9490D for the nominal flight condition.

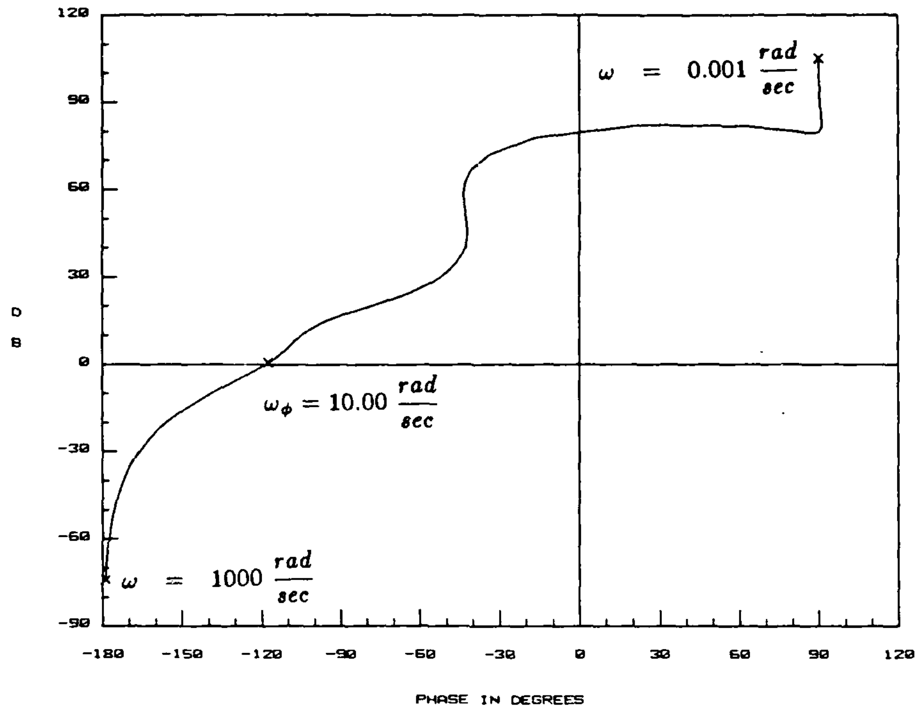


Figure 4.1. Continuous Time Nichols Plot - V vs V_{cmd}

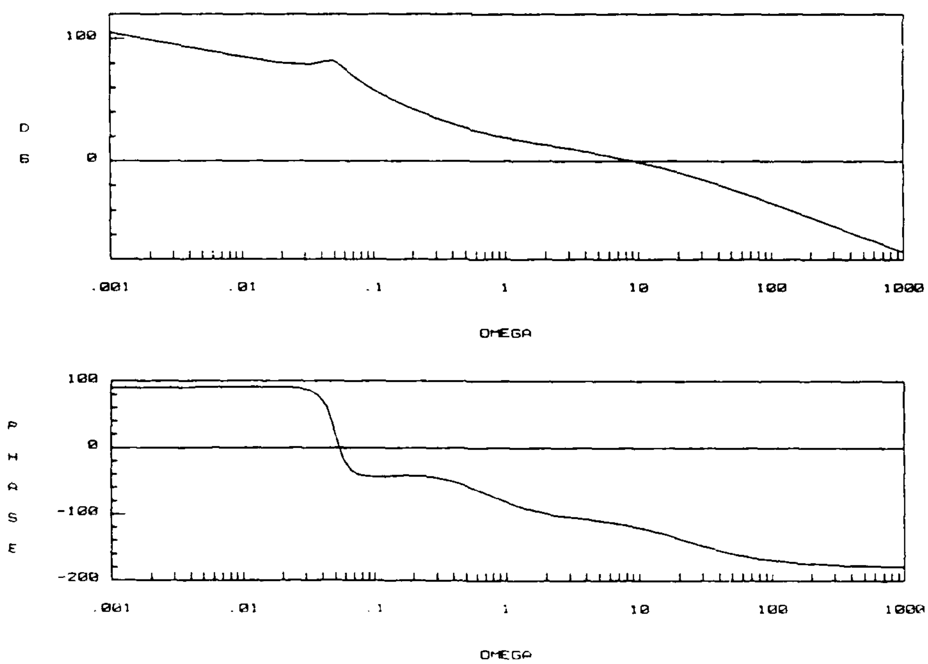


Figure 4.2. Continuous Time Open-Loop Bode Plot - V vs V_{cmd}

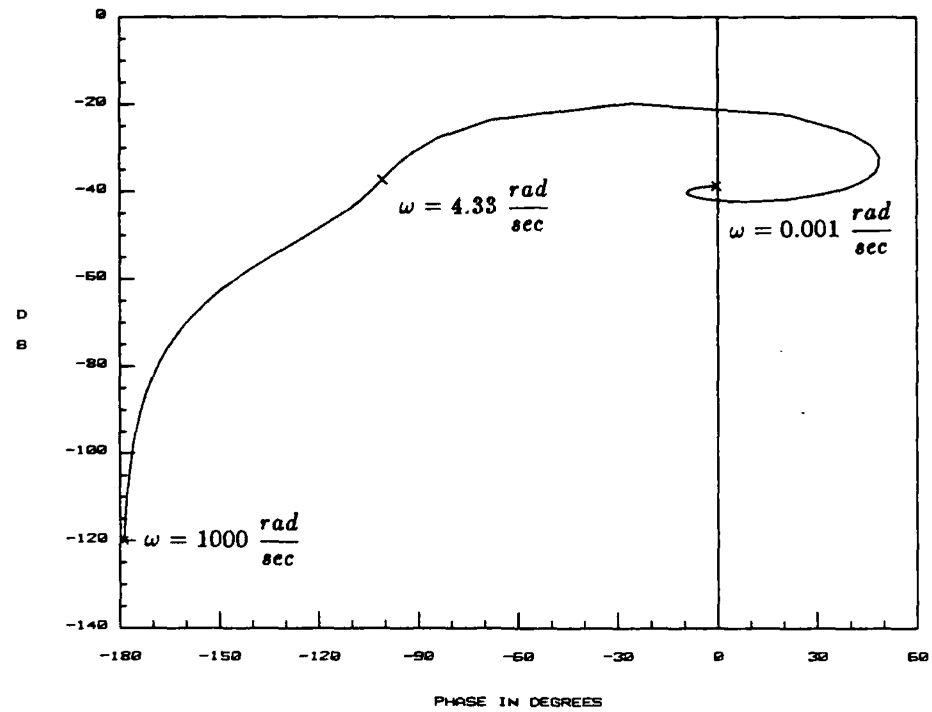


Figure 4.3. Continuous Time Nichols Plot - β vs β_{cmd}

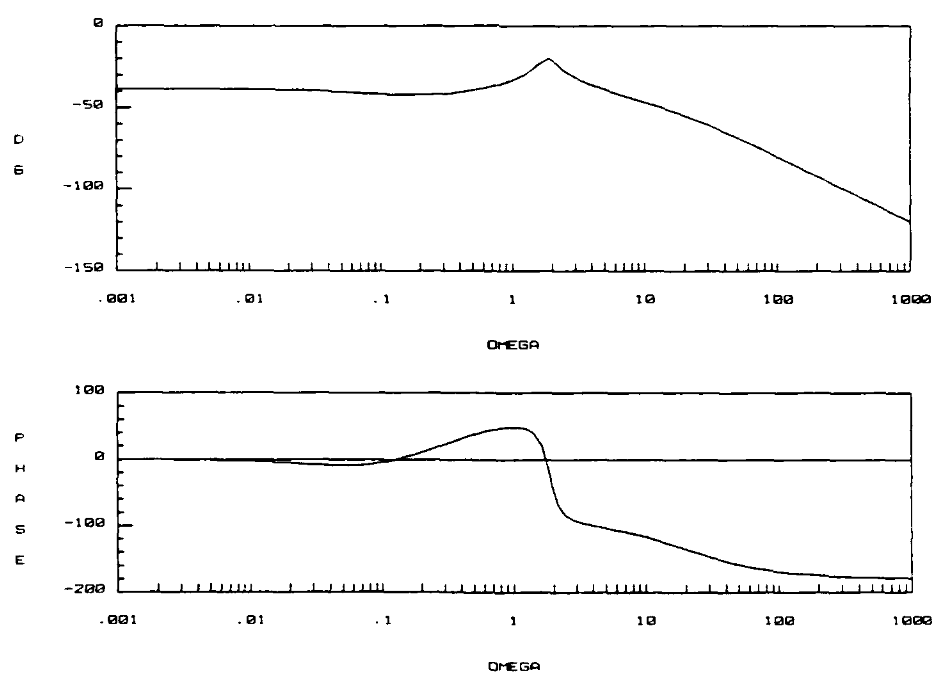


Figure 4.4. Continuous Time Open-Loop Bode Plot - β vs β_{cmd}

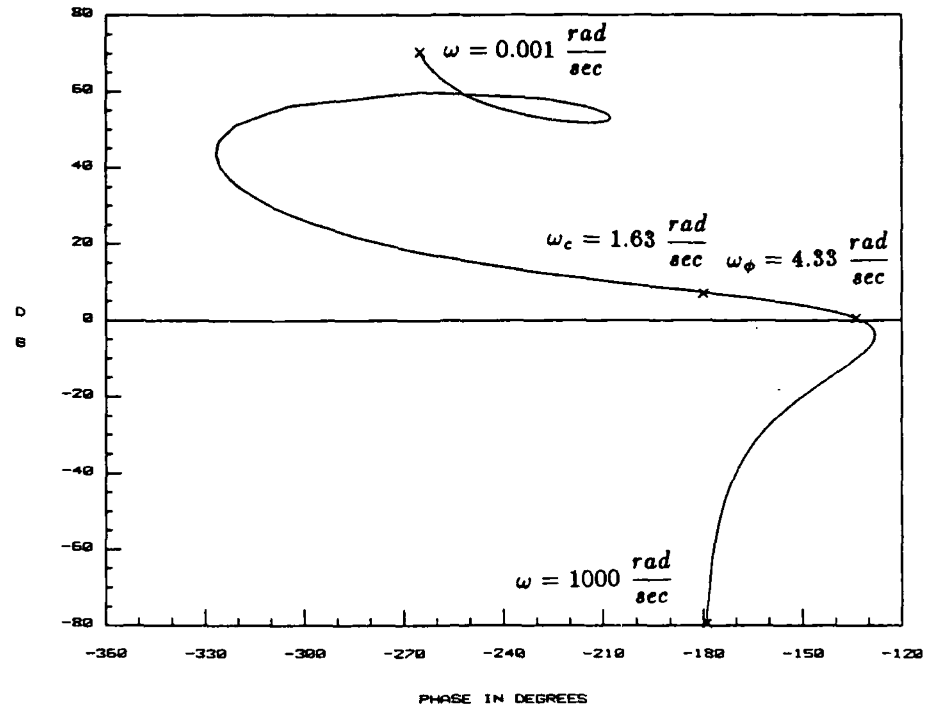


Figure 4.5. Continuous Time Nichols Plot - θ vs θ_{cmd}

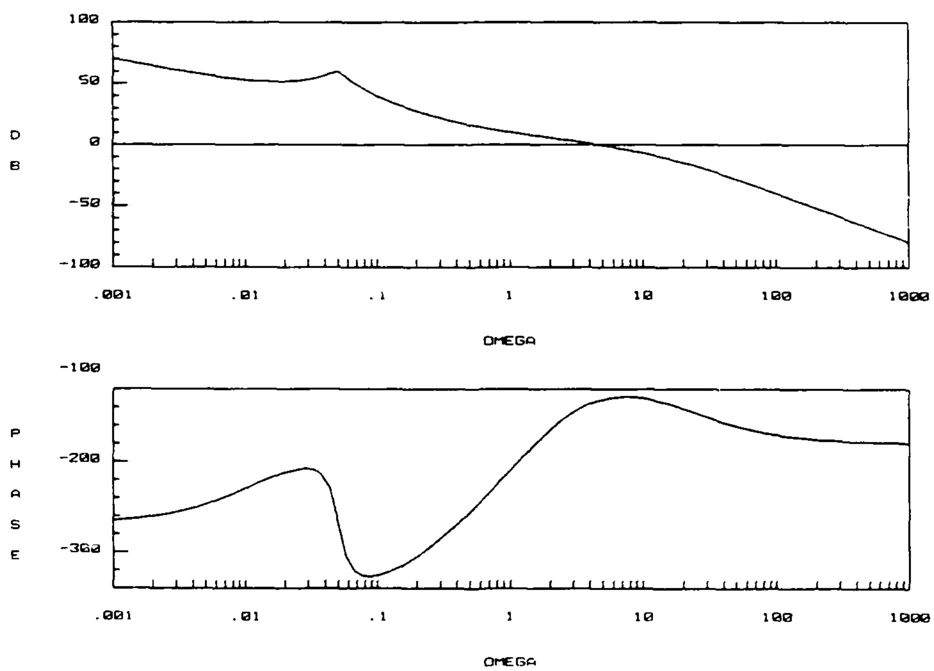


Figure 4.6. Continuous Time Open-Loop Bode Plot θ vs θ_{cmd}

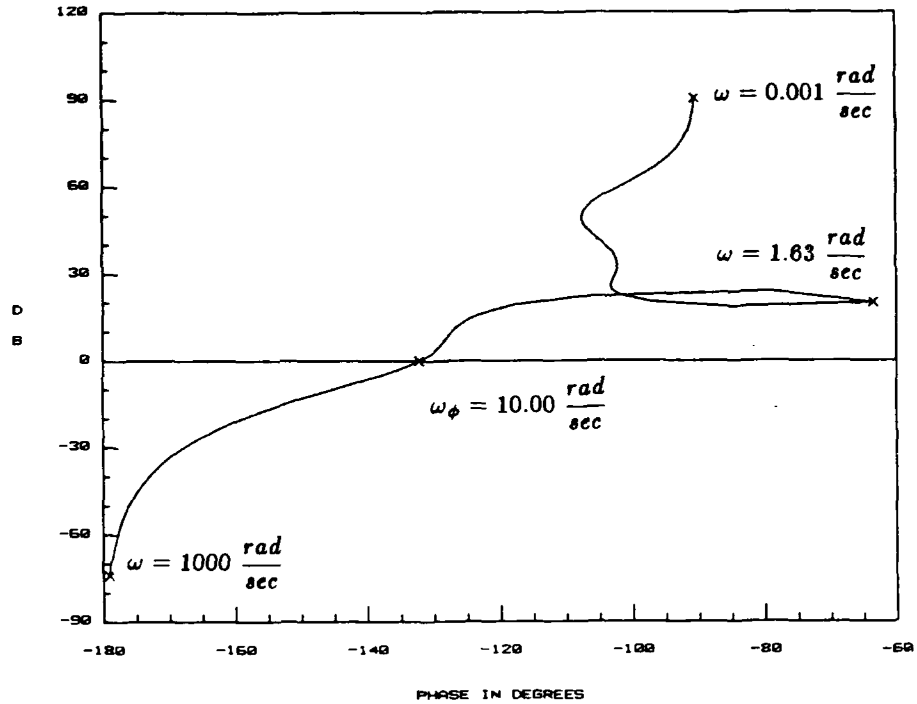


Figure 4.7. Continuous Time Nichols Plot - ϕ vs ϕ_{cmd}

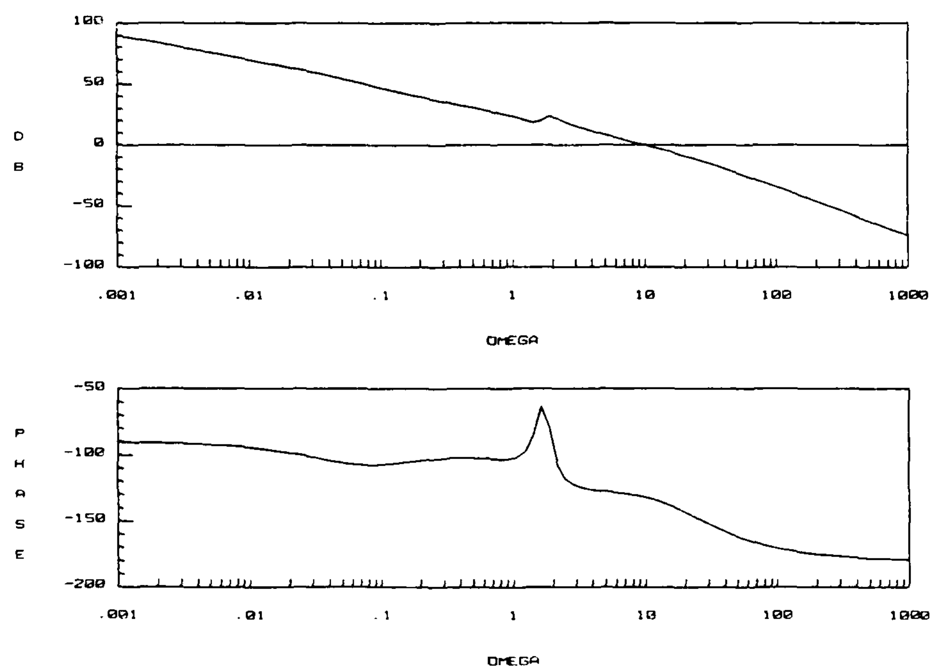


Figure 4.8. Continuous Time Open-Loop Bode Plot ϕ vs ϕ_{cmd}

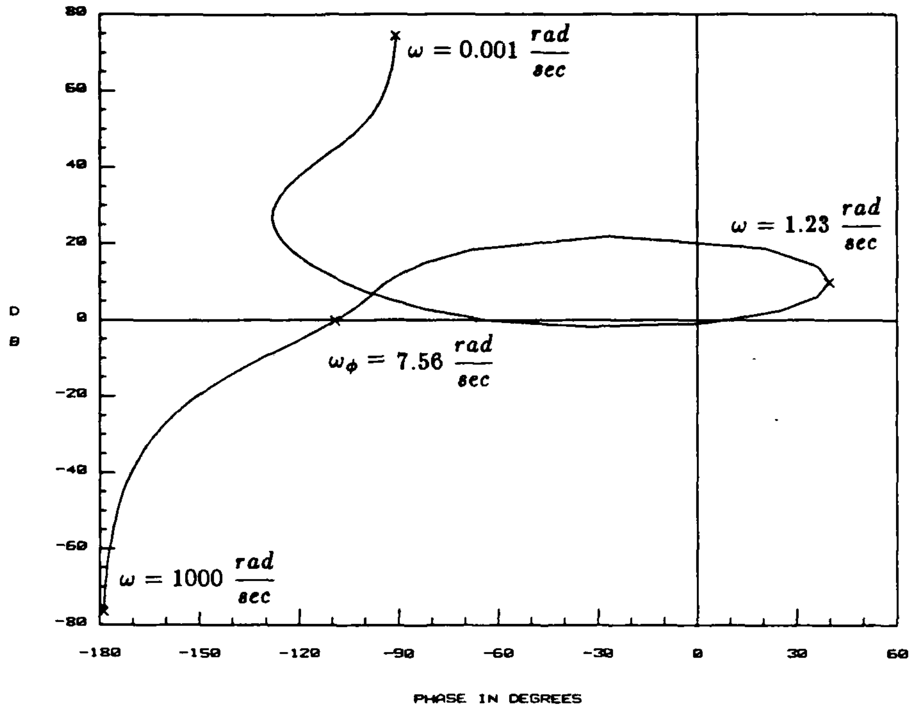


Figure 4.9. Continuous Time Nichols Plot - r vs r_{cmd}

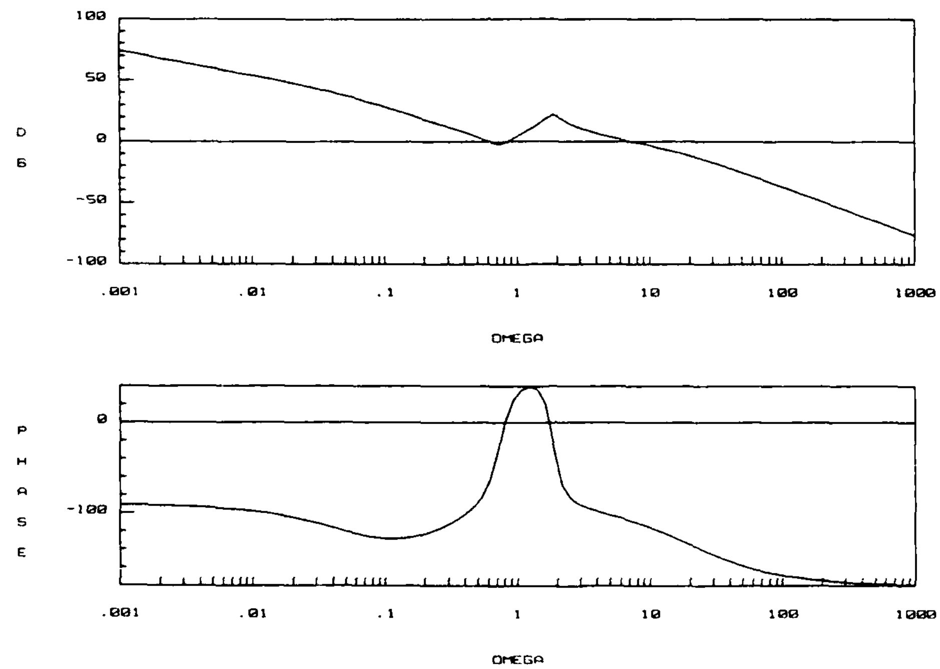


Figure 4.10. Continuous Time Open-Loop Bode Plot - r vs r_{cmd}

The gain and phase margins for ACM Entry and TF/TA are illustrated in Table 4.9.

4.9.1 Asymptotic Transfer Function When the complete system has been simulated and the final design variables selected, the asymptotic transfer functions of Equation 3.23 and Equation 3.24 are determined. The slow transfer function is expressed as

$$\Gamma_s(\lambda) = [C_1 - C_2F_2^{-1}F_1][\lambda I_p - A_{11} + A_{12}F_2^{-1}F_1]^{-1}A_{12}F_2^{-1}$$

$$\Gamma_s(\lambda) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda + 2.5 & 0 \\ 0 & \lambda + 5 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 2.5 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix} \tag{4.34}$$

$$\Gamma_s(\lambda) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2.5}{\lambda+2.5} & 0 & 0 \\ 0 & 0 & 0 & \frac{5.0}{\lambda+5.0} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.35}$$

and contains only the transmission zeros.

The fast transfer function, Equation 3.24, is expressed as

$$\Gamma_f(\lambda) = C_2F_2^{-1}[\lambda I_{n-p} - gF_2B_2K_1]^{-1}gF_2B_2K_1$$

Table 4.9. Gain and Phase Margins - Continuous Time System

ACM Entry

Transfer Function	Gain Margin (dB) Low	Gain Margin (dB) High	Gain Margin ω_c (rad/sec)	Phase Margin (deg)	Phase Margin ω_ϕ (rad/sec)
V	(3)	49.77	244.71	59.77	10.00
β	(3)	95.78	80.00	(1)	(1)
θ	-7.03	(2)	1.63	46.24	4.33
ϕ	(3)	47.34	215.44	47.85	10.00
r	(3)	50.0	215.00	55.00	7.56

TF/TA

Transfer Function	Gain Margin (dB) Low	Gain Margin (dB) High	Gain Margin ω_c (rad/sec)	Phase Margin (deg)	Phase Margin ω_ϕ (rad/sec)
V	-49.50	(2)	.11	61.52	8.70
β	(3)	95.78	80.00	(1)	(1)
θ	-41.56	(2)	.05	100.85	5.72
ϕ	(3)	49.76	247.71	68.27	8.70
r	(3)	43.74	247.71	57.36	15.20

(1) = Response is always less than 0 dB

(2) = No high frequency gain margin (gain margin > 10 dB) because high frequency plot is asymptotic to -180 degrees

(3) = No low frequency gain margin (phase > -180 degrees)

$$
\Gamma_f(\lambda) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda + 10 & 0 & 0 & 0 & 0 \\ 0 & \lambda + .05 & 0 & 0 & 0 \\ 0 & 0 & \lambda + 15 & 0 & 0 \\ 0 & 0 & 0 & \lambda + 10 & 0 \\ 0 & 0 & 0 & 0 & \lambda + 20 \end{bmatrix}^{-1} \bullet
$$

$$
\begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & .05 & 0 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 20 \end{bmatrix}
$$

$$
= \begin{bmatrix} \frac{10}{\lambda+10} & 0 & 0 & 0 & 0 \\ 0 & \frac{.05}{\lambda+.05} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{20}{\lambda+20} \end{bmatrix} \tag{4.36}
$$

The asymptotic transfer functions for all the flight conditions containing the same weighting matrix Σ, measurement matrix M and gain g of Figure 4.7, are the same. Table 4.10 lists the values for the ACM Exit flight condition.

4.4 Discrete Controller Design Variables

The design methodology for the discrete PI controller parallels the continuous system development. That is, the controller gain matrices K_1 and K_2 are described by Equations 3.26 and 3.27

$$
K_1 = (F_2 B_2)^{-1} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_5 \end{bmatrix}
$$

Table 4.10. Asymptotic Transfer Function - ACMEXIT

$$\Gamma_s(\lambda) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{4.0}{\lambda+4.0} & 0 & 0 \\ 0 & 0 & 0 & \frac{4.0}{\lambda+4.0} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma_f(\lambda) = \begin{bmatrix} \frac{5}{\lambda+5} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\lambda+1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{4}{\lambda+4} \end{bmatrix}$$

and

$$K_2 = \lambda K_1$$

However, the discrete PI controller of Figure 3.3 incorporates a fixed gain of 1/T in the forward loop instead of the variable gain "g" indicated in Figure 3.2. Often an acceptable response is obtained by reducing or increasing each Σ weighting element an amount proportional to the difference between "g" in the continuous controller and the sampling frequency 1/T gain factor in the discrete controller. The discrete PI controller in this thesis uses a sampling time of 40 Hz or a period of 0.025 seconds. The initial values of the Σ weighting matrix were the continuous time design variables of Table 4.7 reduced by 1/8 to compensate for the increase in overall system gain, $\frac{40\ Hz}{5(gain)}$. The adjustments of the weighting matrix elements of Σ follow the same guidelines developed in the previous section. Fine tuning of the response results in the weighting matrix and design parameters of Table 4.11.

The discrete PI controller gain matrices for the ACM Entry flight condition are listed in Table 4.12. The open-loop discrete frequency characteristics of the system are analyzed using functions resident in MATRIX_x [15]. The discrete

Table 4.11. Discrete Time System Design Parameters - ACMENTRY

Flight Condition	σ_1	σ_2	σ_3	σ_4	σ_5	λ	m_{31}	m_{42}
ACMENTRY	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20

Table 4.12. Discrete Gain Matrices - ACMENTRY

$$K_1 = \begin{bmatrix} 2.7500D-1 & 2.6579D+0 & 5.6542D+0 & 4.6737D-1 & 3.2537D+0 \\ 2.7500D-1 & -2.6579D+0 & 5.6542D+0 & -4.6737D-1 & -3.2537D+0 \\ 3.9893D-1 & -8.6397D-1 & -7.3916D-1 & 1.2146D+0 & -1.1562D-1 \\ 3.9893D-1 & 8.6397D-1 & -7.3916D-1 & -1.2146D+0 & 1.1562D-1 \\ 0.0000D+0 & 4.5235D+0 & 0.0000D+0 & 1.1521D+0 & -3.4078D-1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1.3750D-3 & 1.3289D-2 & 2.8271D-2 & 2.3368D-3 & 1.6268D-2 \\ 1.3750D-3 & -1.3289D-2 & 2.8271D-2 & -2.3368D-3 & -1.6268D-2 \\ 1.9946D-3 & -4.3199D-3 & -3.6958D-3 & 6.0731D-3 & -5.7808D-4 \\ 1.9946D-3 & 4.3199D-3 & -3.6958D-3 & -6.0731D-3 & 5.7808D-4 \\ 0.0000D-0 & 2.2617D-2 & 0.0000D-1 & 5.7606D-3 & -1.7039D-2 \end{bmatrix}$$

system that is equivalent to a continuous-time system with a zero-order hold and sampler is transformed into the w' plane using a software routine developed by Hofmann [10]. The w' prime plant has the desirable property that $w' \rightarrow s$ as $T \rightarrow 0$ is achieved [11]. This w' plane property establishes the basis for defining a quantity in the w' plane which is analogous to a quantity in the s domain [11, 217]. The transformation from the z to w' plane is as follows:

$$z = \frac{Tw' + 2}{-Tw' + 2} \tag{4.37}$$

Figure 4.11 through Figure 4.20 show the characteristics of the discrete system for ACM Entry and Table 4.13 summarizes the gain and phase margins for the discrete system. Comparison with Table 4.9 shows the discrete PI controller possessing the desirable characteristics of good gain and phase margin.

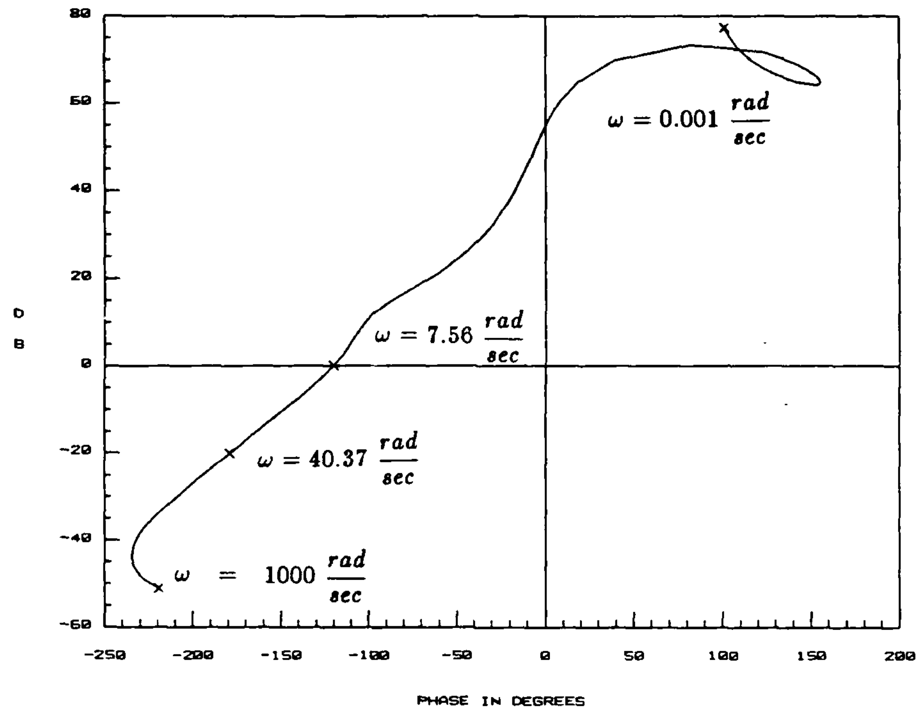


Figure 4.11. Discrete Time Nichols Plot - V vs V_{cmd}

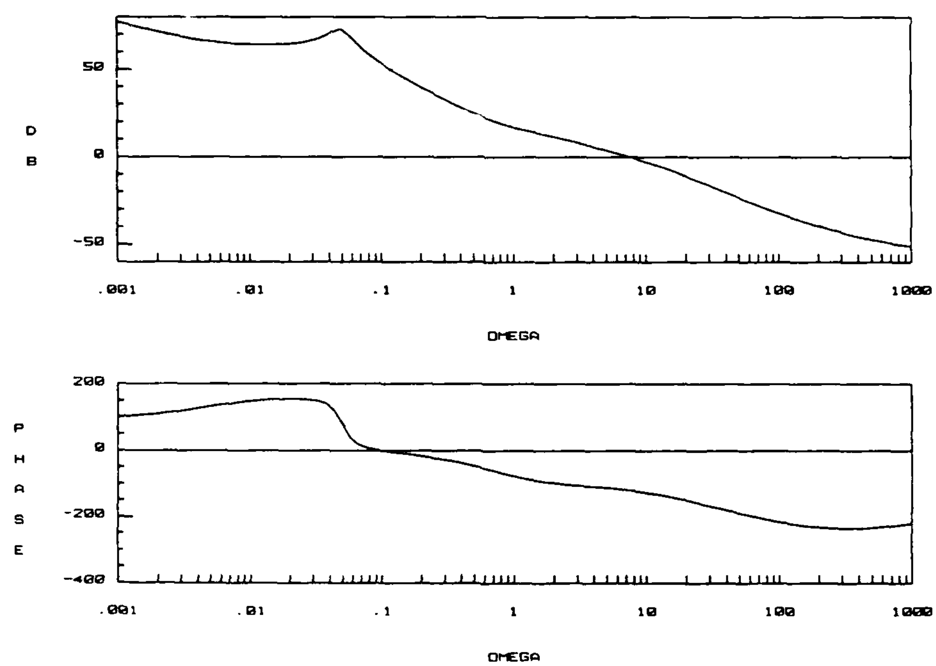


Figure 4.12. Discrete Time Open-Loop Bode Plot - V vs V_{cmd}

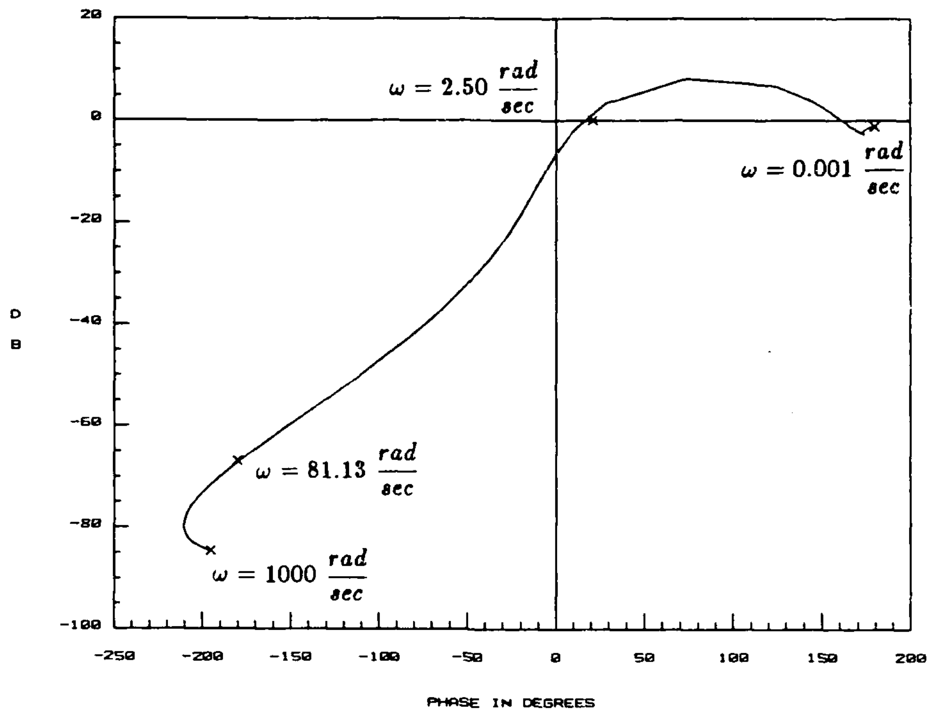


Figure 4.13. Discrete Time Nichols Plot - β vs β_{cmd}

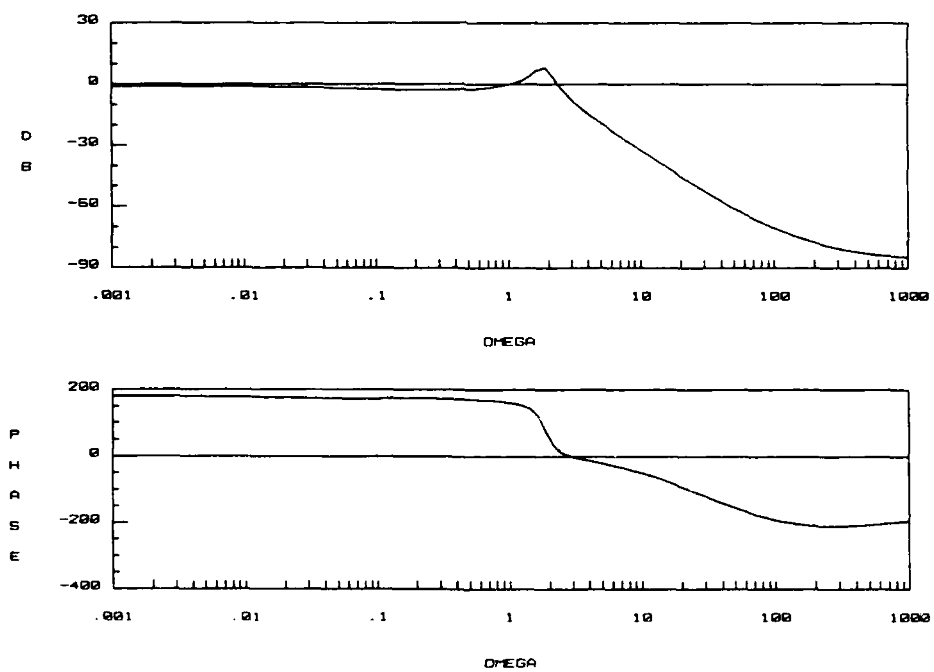


Figure 4.14. Discrete Time Open-Loop Bode Plot - β vs β_{cmd}

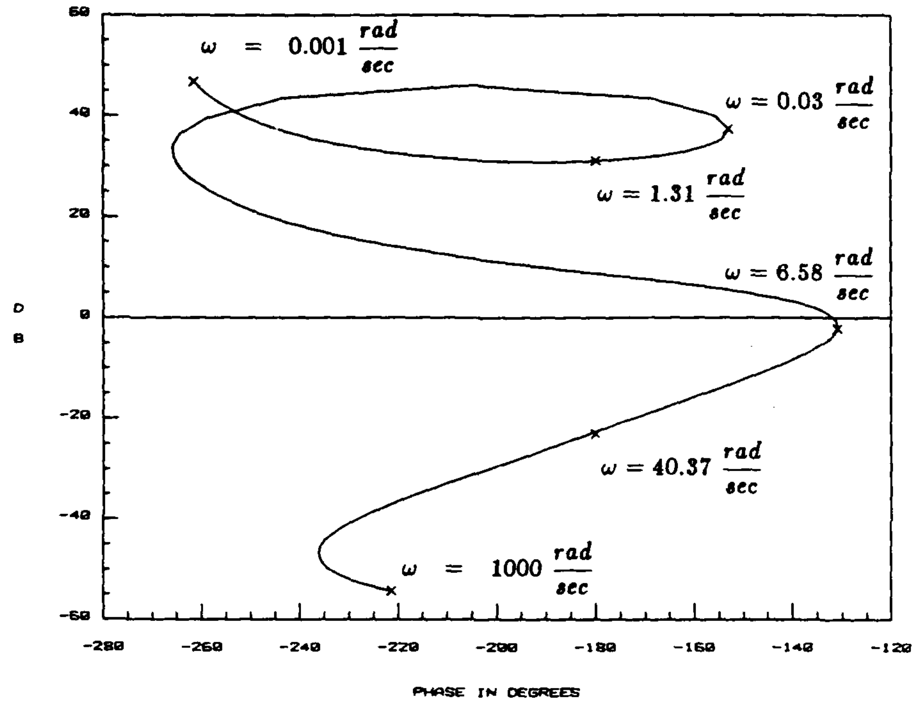


Figure 4.15. Discrete Time Nichols Plot - θ vs θ_{cmd}

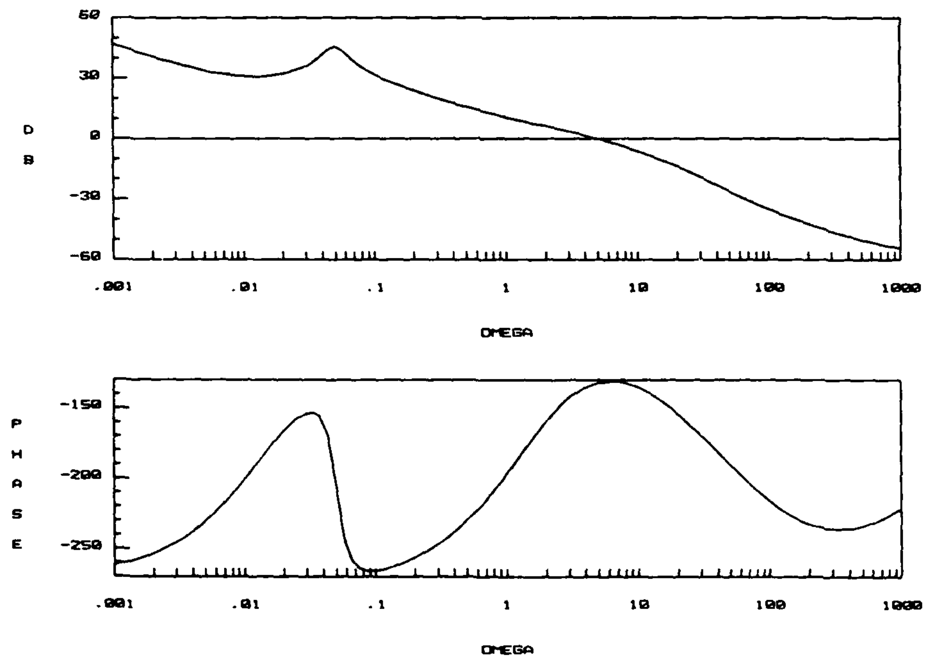


Figure 4.16. Discrete Time Open-Loop Bode Plot θ vs θ_{cmd}

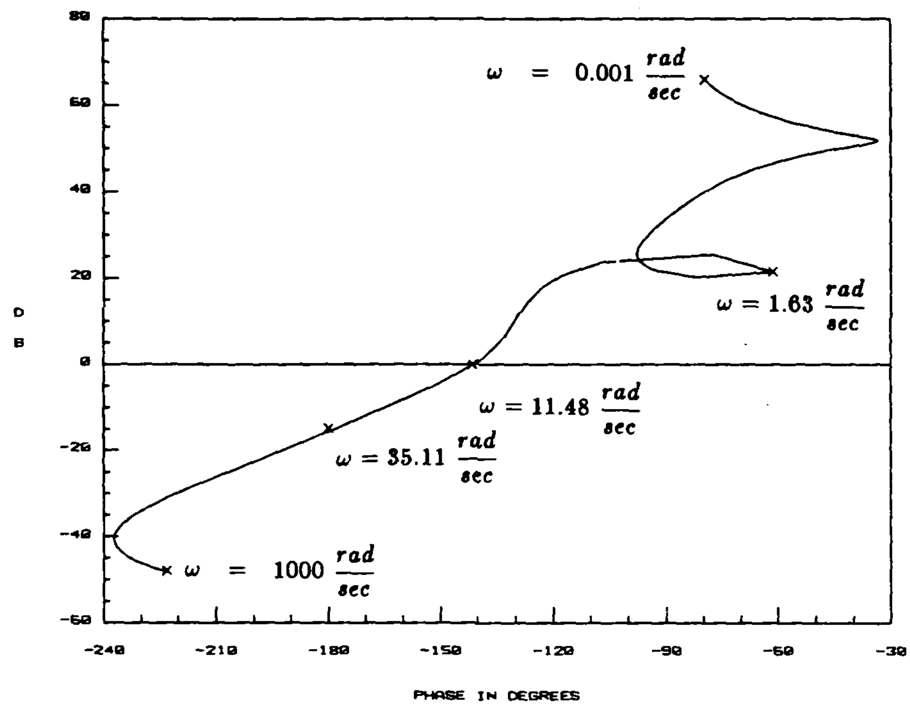


Figure 4.17. Discrete Time Nichols Plot - ϕ vs ϕ_{cmd}

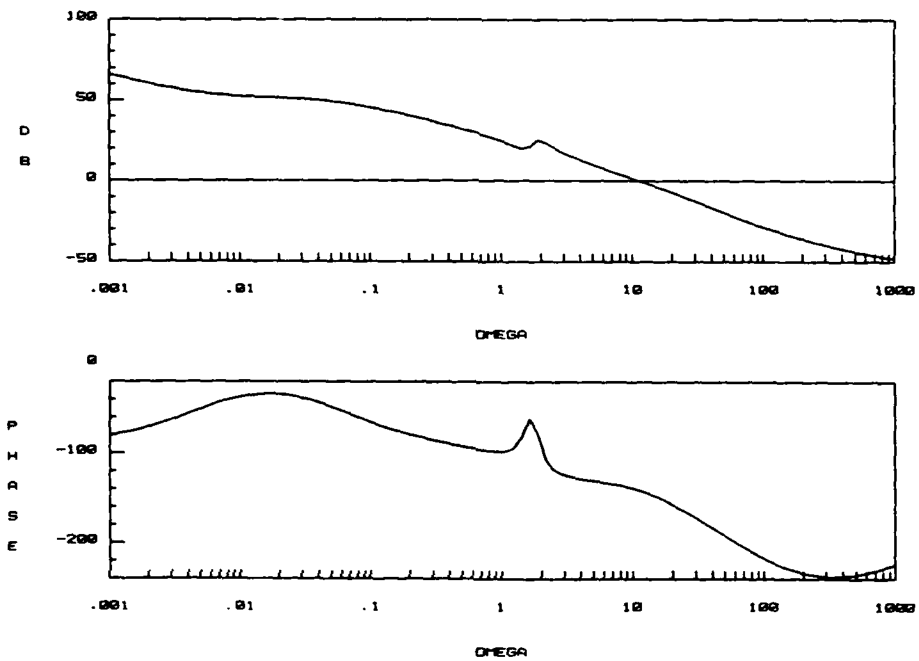


Figure 4.18. Discrete Time Open-Loop Bode Plot ϕ vs ϕ_{cmd}

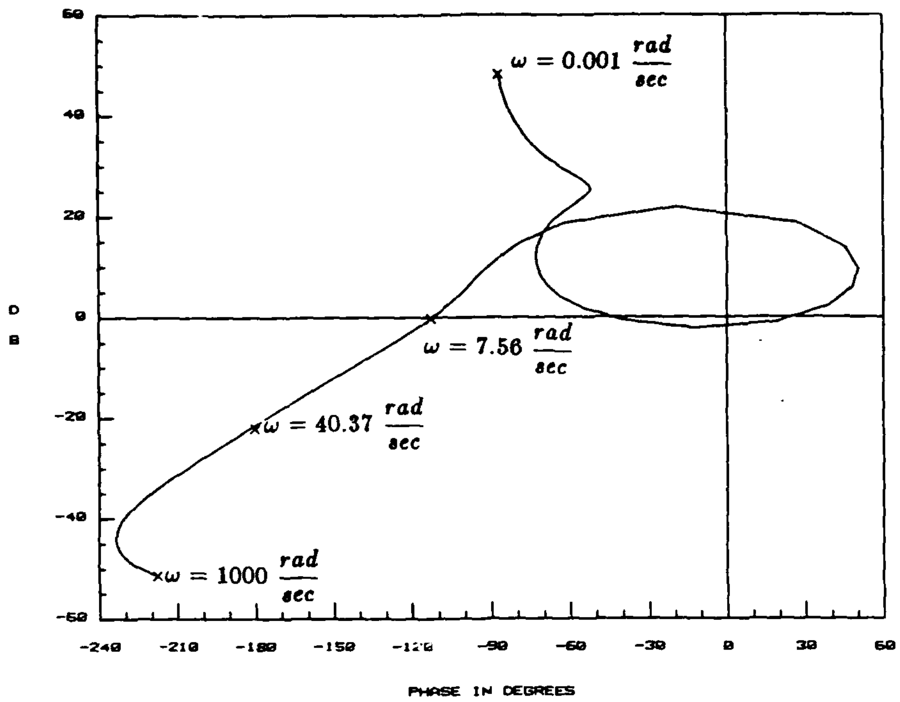


Figure 4.19. Discrete Time Nichols Plot - r vs r_{cmd}

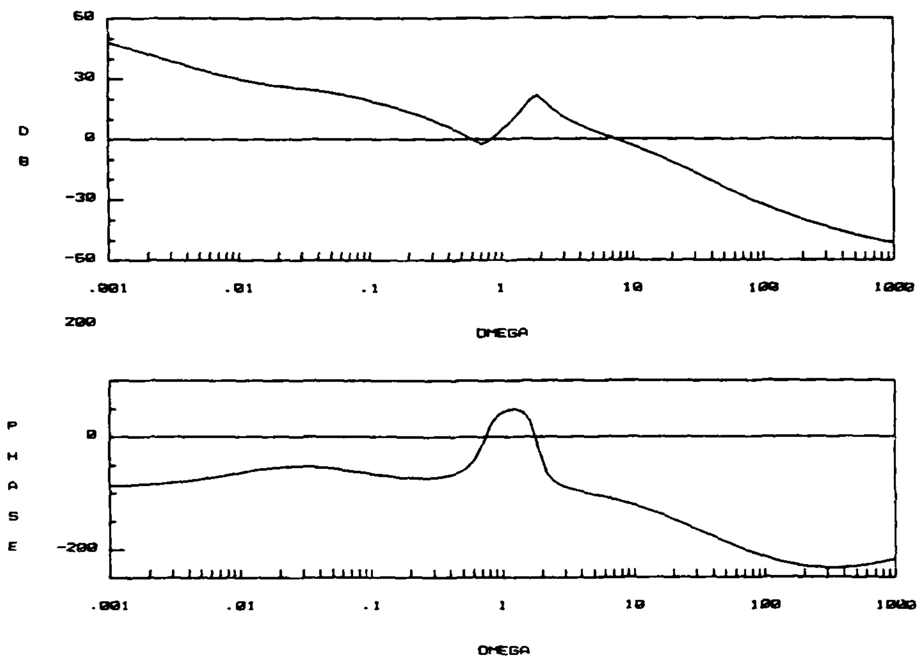


Figure 4.20. Discrete Time Open-Loop Bode Plot - r vs r_{cmd}

The gain and phase margins for ACM Entry and TF/TA are illustrated in Table 4.13. In all the cases where a low frequency gain margin is indicated, the frequency of occurence is 0.01 rad/sec.

4.5 Discrete Controller Design Variables Using Step-Response Matrices

The discrete PI controller implementation using the step-response matrices of Equation 3.28

$$H(T) = \int_0^T C exp(At) B \mathrm{d}t$$

to calculate the gain matrices K_1 and K_2 of Equation 3.31 and Equation 3.32,

$$\begin{aligned} \bar{K}_1 &= H(T)^T[H(T)H(T)^T]^{-1}\Sigma \\ \bar{K}_2 &= G(0)^T[G(0)G(0)^T]^{-1}\Pi \\ \Sigma &= \text{diagonal weighting matrix } [\sigma_1, \ldots \sigma_2] \\ \Pi &= \text{diagonal weighting matrix } [\pi_1, \ldots \pi_m] \end{aligned}$$

uses the same configuration of Figure 3.3. However, due to the small sampling period approximation of H(T) $\approx$ T$[CB]$, the 1/T gain factor in the forward loop is incorporated into $\bar{K}_1$ and $\bar{K}_2$. H(T) and G(0) are calculated using Equation 3.35 and Equation 3.36 from the coefficients of the ARMA difference equation,

$$\begin{aligned} H(T) &= B_1 \\ G(0) &= (I_{(5x5)} + A_1 + A_2)^{-1}(B_1 + B_2) \end{aligned}$$

for the autoregressive moving average (ARMA) representation of the plant. Appendix B gives a procedure to calculate the ARMA model based on constructability invariants. Selecting suitable weighting matrices Σ and Π is an important step prior to implementing the adaptive PI controller of Chapter 6. The step-response design variables for the ACM Entry flight condition are listed in Table 4.14. The step-response PI controller gain matrices for the ACM Entry flight condition are

Table 4.13. Gain and Phase Margins - Discrete Time System

ACM Entry

Transfer Function	Gain Margin (dB)		Gain Margin	Phase Margin	Phase Margin
	Low	High	ω_c (rad/sec)	(deg)	ω_ϕ (rad/sec)
V	(3)	20.26	40.37	60.13	7.56
β	(3)	66.00	81.13	161.61	2.50
θ	-8.76	23.21	40.37	48.16	6.58
ϕ	(3)	14.71	35.11	41.00	11.48
r	(3)	22.16	40.37	68.00	7.56

TF/TA

Transfer Function	Gain Margin (dB)		Gain Margin	Phase Margin	Phase Margin
	Low	High	ω_c (rad/sec)	(deg)	ω_ϕ (rad/sec)
V	-64.55	20.30	40.37	56.46	6.57
β	(3)	17.30	.03	(1)	(1)
θ	-30.31	25.00	40.37	100.28	5.72
ϕ	(3)	16.75	40.37	54.07	11.49
r	(3)	12.26	40.37	47.36	17.47

(1) = Response is always less than 0 dB

(2) = No high frequency gain margin (gain margin > 10 dB) because high frequency plot is asymptotic to -180 degrees

(3) = No low frequency gain margin (phase > -180 degrees)

Table 4.14. Discrete Time System Design Parameters (Step-Response Matrix)

Flight Condition	σ_1	σ_2	σ_3	σ_4	σ_5	$\pi_1 - \pi_5$	m_{31}	m_{42}
ACMENTRY	0.1	0.0015	0.02	0.04	0.6	0.04	0.40	0.20

Table 4.15. Discrete Gain Matrices - ACMENTRY - (Step-Response Matrix)

$$\bar{K}_1 = \begin{bmatrix} 5.4705D+0 & 3.1920D+1 & 2.9162D+2 & 1.0949D+1 & 3.1691D+2 \\ 5.4705D+0 & -3.1920D+1 & 2.9162D+2 & -1.0949D+1 & -3.1691D+2 \\ 7.9174D+0 & -1.0367D+1 & -3.6901D+1 & 1.1511D+1 & -5.7625D+1 \\ 7.9174D+0 & 1.0367D+1 & -3.6901D+1 & -1.1511D+2 & 5.7625D+1 \\ 0.0000D+0 & 5.4361D+1 & 0.0000D+0 & 4.8890D+1 & 1.0760D+2 \end{bmatrix}$$

$$\bar{K}_2 = \begin{bmatrix} 5.6301D-4 & 1.6810D+0 & 1.3864D+0 & -7.6547D-1 & 2.1383D+1 \\ 5.6301D-4 & -1.6810D+0 & 1.3864D+0 & 7.6547D-1 & -2.1383D+1 \\ 5.4982D-4 & 2.7851D-1 & 1.5854D+0 & 2.4882D-1 & -6.9378D+0 \\ 5.4982D-4 & -2.7851D-1 & 1.5854D+0 & -2.4882D-1 & 6.9378D+0 \\ 0.0000D-0 & 4.8837D+0 & 0.0000D+0 & -1.3028D+0 & -3.6074D+1 \end{bmatrix}$$

listed in Table 4.15. The open-loop discrete frequency characteristics of the system have been analyzed using functions resident in MATRIX_x and indicate similar gain and phase margins as the discrete case alone. Table 4.16 summarizes the gain and phase margins applicable to the gain calculations using the step-response matrix method. The low frequency gain margin for the $\frac{\theta}{\theta_{cmd}}$ occurs at 1.32 rad/sec.

4.6 Summary

This chapter presents the process by which the PI control law is implemented for the CRCA. Investigation of the system's controllability, observability, transmission zeros and open-loop eigenvalues reveal essential information that is needed prior to system simulation. The plant must be controllable and observable to implement this type of controller and the system transmission zeros should lie in the left half S-plane to insure a completely stable system. Static instability of the system is indicated from the open-loop eigenvalues of the system. Consequently, the characteristics of low and high frequency gain margins must be considered when controller parameters are adjusted for the desired output response. Examining the control surface deflection values needed to sustain the desired output provides

Table 4.16. Gain and Phase Margins - Step-Response Matrix System

ACM Entry

Transfer Function	Gain Margin (dB) Low	Gain Margin (dB) High	Gain Margin ω_c (rad/sec)	Phase Margin (deg)	Phase Margin ω_ϕ (rad/sec)
V	(3)	26.35	40.37	72.22	3.76
β	(3)	19.82	1.07	87.95	.04
θ	-9.48	22.61	40.37	48.18	4.98
ϕ	(3)	14.62	35.11	41.40	10.00
r	(3)	35.18	40.37	87.51	2.85

TF/TA

Transfer Function	Gain Margin (dB) Low	Gain Margin (dB) High	Gain Margin ω_c (rad/sec)	Phase Margin (deg)	Phase Margin ω_ϕ (rad/sec)
V	(3)	26.33	40.37	87.51	2.85
β	(3)	62.72	40.37	89.96	.04
θ	(3)	25.54	40.37	113.28	4.97
ϕ	(3)	11.83	40.37	43.92	15.92
r	(3)	18.02	40.37	63.28	10.00

(1) = Response is always less than 0 dB

(2) = No high frequency gain margin (gain margin > 10 dB) because high frequency plot is asymptotic to -180 degrees

(3) = No low frequency gain margin (phase > -180 degrees)

insight into plant limitations prior to controller design. The selection of PI controller gains is aided by using the frequency domain design techniques and analysis of Nichols and Bode plots. The simulation results for the design parameters selected are presented in Chapter 5.

V. Simulation Results

5.1 Introduction

This chapter presents a listing of the design parameters and final gains K_1 and K_2 for all flight conditions, including surface failures, and a sample of time response graphs used to evaluate the selection of the final controller gains for the ACM Entry flight condition. The selection of the PI controller design parameters is based on many hundreds of simulations with MATRIX_x [15]. The System Build capability within MATRIX_x allows the controller and plant to be symbolized by the familiar block diagram representation of Figures 3.2 and 3.3. Furthermore, the designer can easy modify individual blocks during the simulation as adjustments are made. The MATRIX_x graphs presented are organized to show the minimal cross coupling that exists between the five input commands and the five outputs, a highly desirable feature of the design technique. Two basic types of input commands are evaluated. The first is a ramped step input command and the other is a model following input designed to provide a third order response for the coordinated turn and pitch tracking tasks. The control surface deflections, rates, and state variable outputs for ACM Entry are contained in Appendix C. The responses for the remaining cases, Table 4.1, are available under separate cover. However, the PI controller gains K_1 and K_2 for *all* cases are included.

5.2 Controller Design Variables

Ultimately, the multivariable control engineer seeks to find a set of PI controller gains, K_1 and K_2, that work satisfactorily over the entire flight envelope. Because of altitude, speed, and aircraft changes, the universal gain concept cannot provide good responses at all set points. However, gain calculations made at a high $\bar{q}$ condition (high dynamic pressure) often work at many other points in the flight envelope. In this research effort, the PI control law gains calculated at the TF/TA

Table 5.1. Stability Analysis Using TF/TA PI Controller Gains

Flight Condition	Stable With Universal Gain	Requires Gain Change
ACMENTRY	X	
ACM30TL		X
ACM50CL	X	
ACMEXIT		X
TFTA	X	
TFTA30TL	X	
TFTA50CL	X	

where,

ACMENTRY = ACM Entry(nominal flight condition)
ACM30TL = 30 percent loss of effectiveness of the left trailing edge - ACM Entry
ACM50CL = 50 percent loss of effectiveness of the left canard - ACM Entry
ACMEXIT = ACM Exit flight condition
TFTA = TF/TA flight condition
TFTA30TL = 30 percent loss of effectiveness of the left trailing edge -TF/TA
TFTA50CL = 50 percent loss of effectiveness of the left canard - TF/TA

flight condition provide stable responses for 70 percent of the cases considered. Table 5.1 displays those flight conditions. For the fixed gain PI controller, the final selection of the controller design parameters are made in a effort to find weighting matrices, Σ and Π, proportionality constant λ, and measurement matrix M, that work well at all set points selected. This design philosophy allows simplified gain calculating and scheduling by minimizing the storage of controller variables. Table 5.2, Table 5.10, and Table 5.18, tabulate the design parameter selections for the continuous, discrete, and step-response matrix methods, respectively. The continuous time controller gains, K_1 and K_2 are also listed, beginning with Table 5.3. Table 5.11 and 5.19 begin the gain presentations for the two discrete controller designs.

Table 5.2. Continuous Time System Design Parameters

Flight Condition	σ_1	σ_2	σ_3	σ_4	σ_5	λ	g	m_{31}	m_{42}
ACMENTRY	2.0	0.01	3.0	2.0	4.0	0.1	5.0	0.40	0.20
ACM30TL	2.0	0.01	3.0	2.0	4.0	0.1	5.0	0.40	0.20
ACM50CL	2.0	0.01	3.0	2.0	4.0	0.1	5.0	0.40	0.20
ACMEXIT	1.0	0.20	1.1	2.0	0.8	0.1	5.0	0.25	0.25
TFTA	2.0	0.01	1.1	2.0	4.0	0.1	5.0	0.40	0.20
TFTA30TL	2.0	0.01	1.1	2.0	4.0	0.1	5.0	0.40	0.20
TFTA50CL	2.0	0.01	1.1	2.0	4.0	0.1	5.0	0.40	0.20

Table 5.3. Continuous PI Controller Gain Matrices - ACMENTRY

$$K_1 = \begin{bmatrix} 2.7500D+0 & 5.3158D+0 & 3.3925D-1 & 3.1158D-0 & 2.6029D-1 \\ 2.7500D+0 & 5.3158D+0 & 3.3925D-1 & -3.1158D-0 & -2.6029D-1 \\ 3.9893D+0 & -1.7279D+0 & -4.4350D-0 & 8.0975D-0 & -9.2492D-1 \\ 3.9893D-0 & 1.7279D+0 & -4.4350D-0 & -8.0975D-0 & 9.2492D-1 \\ 0.0000D+0 & 9.0470D+0 & 0.0000D-0 & 7.6808D-0 & -2.7263D-1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 2.7500D-1 & 5.3158D-1 & 3.3925D-0 & 3.1158D-1 & 2.6029D+0 \\ 2.7500D-1 & 5.3158D-1 & 3.3925D-0 & -3.1158D-1 & -2.6029D-0 \\ 3.9893D-1 & -1.7279D-1 & -4.4350D-1 & 8.0975D-1 & -9.2492D-2 \\ 3.9893D-1 & 1.7279D-1 & -4.4350D-1 & -8.0975D-1 & 9.2492D-2 \\ 0.0000D-1 & 9.0470D-1 & 0.0000D-1 & 7.6808D-1 & -2.7263D-0 \end{bmatrix}$$

Table 5.4. Continuous PI Controller Gain Matrices - ACM30TL

$$K_1 = \begin{bmatrix} 3.3161D+0 & -5.6001D-0 & 3.1719D-1 & -2.6372D-0 & 4.2981D-1 \\ 3.4104D-0 & 5.6334D-0 & 3.2308D+1 & 1.3810D-0 & -4.2150D-1 \\ 8.4279D-0 & -3.9333D-1 & -1.2050D+1 & 1.4817D-1 & -9.7984D-0 \\ 4.9356D-0 & 2.8302D-1 & -6.8869D-0 & -1.0661D-1 & 7.0502D-0 \\ 0.0000D-0 & 1.0000D-1 & 0.0000D-0 & 0.0000D-0 & 0.0000D-0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 3.3161D-1 & -5.6001D-1 & 3.1719D-0 & -2.6372D-1 & 4.2981D-0 \\ 3.4104D-1 & 5.6334D-1 & 3.2308D-0 & 1.3810D-1 & -4.2150D-0 \\ 8.4279D-1 & -3.9333D-2 & -1.2050D-0 & 1.4817D-0 & -9.7984D-1 \\ 4.9356D-1 & 2.8302D-2 & -6.8869D-1 & -1.0661D-0 & 7.0502D-1 \\ 0.0000D-0 & 1.0000D-0 & 0.0000D-0 & 0.0000D-0 & 0.0000D-0 \end{bmatrix}$$

Table 5.5. Continuous PI Controller Gain Matrices - ACM50CL

$$K_1 = \begin{bmatrix} -1.6235D+2 & -5.2382D+1 & -2.0460D+3 & 1.3838D+1 & -3.6970D-2 \\ -3.6411D+1 & -1.7165D+1 & -4.5886D+2 & 4.5348D+0 & -1.2115D-2 \\ -5.3619D+1 & -2.0925D+1 & -7.2928D+2 & 1.4609D+1 & -1.4030D-2 \\ -6.7328D+1 & -2.3670D+1 & -9.0205D+2 & -2.8281D+0 & -1.7445D-2 \\ -3.6411D+1 & -7.1654D+0 & -4.5886D+2 & 4.5348D+0 & -1.2115D-2 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -1.6235D+1 & -5.2382D+0 & -2.0460D+2 & 1.3838D+0 & -3.6970D-1 \\ -3.6411D+0 & -1.7165D+0 & -4.5886D+1 & 4.5348D-1 & -1.2115D-1 \\ -5.3619D+0 & -2.0925D+0 & -7.2928D+1 & 1.4609D+0 & -1.4030D-1 \\ -6.7328D+0 & -2.3670D+0 & -9.0205D+1 & -2.8281D-1 & -1.7445D-1 \\ -3.6411D+0 & -7.1654D-1 & -4.5886D+1 & 4.5348D-1 & -1.2115D-1 \end{bmatrix}$$

Table 5.6. Continuous PI Controller Gain Matrices - ACMEXIT

$$K_1 = \begin{bmatrix} -1.4384D+0 & 7.2554D+1 & 1.0439D-2 & 1.5985D-1 & 1.1507D-1 \\ -1.3641D+0 & -7.4061D+1 & 9.8991D+1 & -1.6317D-1 & -1.1746D-1 \\ -4.0262D+0 & -5.7876D+1 & -8.7124D+1 & 5.2047D-1 & 1.2687D+0 \\ -3.6355D+0 & 6.0078D-1 & -1.1548D-2 & -5.1562D-1 & -9.1953D-1 \\ -1.6292D-1 & 1.5496D-2 & 1.1823D-1 & 2.1180D-1 & -3.9052D-1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -1.4384D-1 & 7.2554D+0 & 1.0439D-1 & 1.5985D+0 & 1.1507D+0 \\ -1.3641D-1 & -7.4061D+0 & 9.8991D+0 & -1.6317D+0 & -1.1746D+0 \\ -4.0262D-1 & -5.7876D+0 & -8.7124D+0 & 5.2047D+0 & 1.2687D-1 \\ -3.6355D-1 & 6.0078D+0 & -1.1548D-1 & -5.1562D+0 & -9.1953D-2 \\ -1.6292D-2 & 1.5496D+1 & 1.1823D+0 & 2.1180D+0 & -3.9052D+0 \end{bmatrix}$$

Table 5.7. Continuous PI Controller Gain Matrices - TFTA

$$K_1 = \begin{bmatrix} 1.3716D+0 & 1.6720D+0 & 2.2989D+1 & 5.8038D-1 & 7.1864D+0 \\ -1.5494D+0 & -1.6720D+0 & 2.3932D+1 & -5.8038D-1 & -7.1864D+0 \\ 2.7067D+0 & -9.3130D-1 & 1.4363D+1 & 4.6092D+0 & 7.4561D-2 \\ 2.6077D+0 & 9.3130D-1 & 1.3837D+1 & -4.6092D+0 & -7.4561D-2 \\ -1.5477D-1 & 2.9123D+0 & -8.2127D-1 & 1.2747D+0 & -9.5488D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1.3716D-1 & 1.6720D-1 & 2.2989D+0 & 5.8038D-2 & 7.1864D-1 \\ -1.5494D-1 & -1.6720D-1 & 2.3932D+0 & -5.8038D-2 & -7.1864D-1 \\ 2.7067D-1 & -9.3130D-2 & 1.4363D+0 & 4.6092D-1 & 7.4561D-3 \\ 2.6077D-1 & 9.3130D-2 & 1.3837D+0 & -4.6092D-1 & -7.4561D-3 \\ -1.5477D-2 & 2.9123D-1 & -8.2127D-2 & 1.2747D-1 & -9.5488D-1 \end{bmatrix}$$

Table 5.8. Continuous PI Controller Gain Matrices - TFTA30TL

$$K_1 = \begin{bmatrix} 1.2984D+0 & 1.9174D+0 & 2.3231D+1 & -5.6253D-1 & 8.1869D+0 \\ 1.2931D+0 & -1.7192D+0 & 2.3201D+1 & -4.0717D-1 & -8.1175D+0 \\ 4.0678D+0 & -1.4479D+0 & 2.2999D+1 & 7.0830D+0 & -5.0695D-1 \\ 2.3628D+0 & 1.2252D+0 & 1.3359D+1 & -5.9934D+0 & 4.2897D-1 \\ -2.6797D-3 & 3.1817D+0 & -1.5151D-2 & 7.7678D-2 & -8.1522D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1.2984D-1 & 1.9174D-1 & 2.3231D+0 & -5.6253D-2 & 8.1869D-1 \\ 1.2931D-1 & -1.7192D-1 & 2.3201D+0 & -4.0717D-2 & -8.1175D-1 \\ 4.0678D-1 & -1.4479D-1 & 2.2999D+0 & 7.0830D-1 & -5.0695D-2 \\ 2.3628D-1 & 1.2252D-1 & 1.3359D+0 & -5.9934D-1 & 4.2897D-2 \\ -2.6797D-4 & 3.1817D-1 & -1.5151D-3 & 7.7678D-3 & -8.1522D-1 \end{bmatrix}$$

Table 5.9. Continuous PI Controller Gain Matrices - TFTA50CL

$$K_1 = \begin{bmatrix} 4.0684D+0 & 4.4327D+0 & 6.6736D+1 & -1.1026D-1 & 1.7040D+1 \\ 2.0452D+0 & -4.4403D-1 & 3.3549D+1 & 1.1045D-2 & -1.7069D+0 \\ 3.7702D+0 & 9.1527D-1 & 3.7010D+1 & 4.2682D+0 & 7.0732D+0 \\ 3.6996D+0 & 2.3565D+0 & 3.5852D+1 & -4.3496D+0 & 5.5039D+0 \\ 1.1003D-2 & 2.3396D+0 & 1.8048D-1 & 6.6177D-2 & -1.0227D+1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 4.0684D-1 & 4.4327D-1 & 6.6736D+0 & -1.1026D-2 & 1.7040D+0 \\ 2.0452D-1 & -4.4403D-2 & 3.3549D+0 & 1.1045D-3 & -1.7069D-1 \\ 3.7702D-1 & 9.1527D-2 & 3.7010D+0 & 4.2682D-1 & 7.0732D-1 \\ 3.6996D-1 & 2.3565D-1 & 3.5852D+0 & -4.3496D-1 & 5.5039D-1 \\ 1.1003D-3 & 2.3396D-1 & 1.8048D-2 & 6.6177D-3 & -1.0227D+0 \end{bmatrix}$$

Table 5.10. Discrete Time System Design Parameters

Flight Condition	σ_1	σ_2	σ_3	σ_4	σ_5	λ	m_{31}	m_{42}
ACMENTRY	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20
ACM30TL	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20
ACM50CL	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20
ACMEXIT	0.1	0.025	0.5	0.3	0.7	0.005	0.25	0.25
TFTA	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20
TFTA30TL	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20
TFTA50CL	0.2	0.005	0.5	0.3	0.5	0.005	0.40	0.20

Table 5.11. Discrete PI Controller Gain Matrices - ACMENTRY

$$K_1 = \begin{bmatrix} 2.7500D-1 & 2.6579D+0 & 5.6542D+0 & 4.6737D-1 & 3.2537D+0 \\ 2.7500D-1 & -2.6579D+0 & 5.6542D+0 & -4.6737D-1 & -3.2537D+0 \\ 3.9893D-1 & -8.6397D-1 & -7.3916D-1 & 1.2146D+0 & -1.1562D-1 \\ 3.9893D-1 & 8.6397D-1 & -7.3916D-1 & -1.2146D+0 & 1.1562D-1 \\ 0.0000D+0 & 4.5235D+0 & 0.00000+0 & 1.1521D+0 & -3.4078D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1.3750D-3 & 1.3289D-2 & 2.8271D-2 & 2.3368D-3 & 1.6268D-2 \\ 1.3750D-3 & -1.3289D-2 & 2.8271D-2 & -2.3368D-3 & -1.6268D-2 \\ 1.9946D-3 & -4.3199D-3 & -3.6958D-3 & 6.0731D-3 & -5.7808D-4 \\ 1.9946D-3 & 4.3199D-3 & -3.6958D-3 & -6.0731D-3 & 5.7808D-4 \\ 0.0000D-0 & 2.2617D-2 & 0.00000-0 & 5.7606D-3 & -1.7039D-2 \end{bmatrix}$$

Table 5.12. Discrete PI Controller Gain Matrices - ACM30TL

$$K_1 = \begin{bmatrix} 3.3161D-1 & -2.8000D+0 & 5.2865D+0 & -3.9558D-1 & 5.3726D+0 \\ 3.4104D-1 & 2.8167D+0 & 5.3847D+0 & 2.0715D-1 & -5.2688D+0 \\ 8.4279D-1 & -1.9667D-1 & -2.0084D+0 & 2.2225D+0 & -1.2248D+0 \\ 4.9356D-1 & 1.4151D-1 & -1.1478D+0 & -1.5992D+0 & 7.0502D-1 \\ 0.0000D+0 & 5.0000D+0 & 0.0000D+0 & 0.0000D+0 & 0.0000D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1.6580D-3 & -1.4000D-2 & 2.6432D-2 & -1.9779D-3 & 2.6863D-2 \\ 1.7052D-3 & 1.4084D-2 & 2.6923D-2 & 1.0358D-3 & -2.6344D-2 \\ 4.2140D-3 & -9.8334D-4 & -1.0042D-2 & 1.1113D-3 & -6.1240D-3 \\ 2.4678D-3 & 7.0754D-4 & -5.7931D-3 & -7.9959D-3 & 4.4064D-3 \\ 0.0000D+0 & 2.5000D-2 & 0.0000D+0 & 0.0000D+0 & 0.0000D+0 \end{bmatrix}$$

Table 5.13. Discrete PI Controller Gain Matrices - ACM50CL

$$K_1 = \begin{bmatrix} -1.6235D+1 & -2.6191D+1 & -3.4100D+2 & 2.0758D+0 & -4.6212D+1 \\ -3.6411D+0 & -8.5827D+0 & -7.6477D+1 & 6.8022D-1 & -1.5144D+1 \\ -5.3619D+0 & -1.0462D+1 & -1.2155D+2 & 2.1914D+0 & -1.7537D+1 \\ -6.7328D+0 & -1.1835D+1 & -1.5034D+2 & -4.2422D+1 & -2.1806D+1 \\ -3.6411D+0 & -3.5827D+0 & -7.6477D+1 & 6.8022D-1 & -1.5144D+1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -8.1176D-2 & -1.3095D-1 & -1.7050D+0 & 1.0379D-2 & -2.3106D-1 \\ -1.8206D-2 & -4.2913D-2 & -3.8238D-1 & 4.4011D-3 & -7.5719D-2 \\ -2.6809D-2 & -5.2312D-2 & -6.0774D-1 & 1.0957D-2 & -8.7686D-2 \\ -3.3664D-2 & -5.9175D-2 & -7.5171D-1 & -2.1211D-3 & -1.0903D-1 \\ -1.8206D-2 & -1.7913D-2 & -3.8238D-1 & 3.4011D-3 & -7.5719D-2 \end{bmatrix}$$

Table 5.14. Discrete PI Controller Gain Matrices - ACMEXIT

$$K_1 = \begin{bmatrix} -1.4384D-1 & 9.0693D+0 & 4.7449D+1 & 2.3977D+0 & 1.0069D+1 \\ -1.3641D-1 & -9.2576D+0 & 4.4996D+1 & -2.4475D+0 & -1.0278D+1 \\ -4.0262D-1 & -7.2345D+0 & -3.9602D+1 & 7.8071D+0 & 1.1101D+0 \\ -3.6355D-1 & 7.5097D+0 & -5.2490D+1 & -7.7343D-0 & -8.0459D-1 \\ -1.6292D-2 & 1.9370D+1 & 5.3741D+0 & 3.1770D-0 & -3.4170D+1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -7.1921D-4 & 4.4346D-2 & 2.3724D-1 & 1.1989D-2 & 5.0343D-2 \\ -6.8203D-4 & -4.6288D-2 & 2.2498D-1 & -1.2238D-2 & -5.1389D-2 \\ -2.0131D-3 & -3.6173D-2 & -1.9801D-1 & 3.9036D-2 & 5.5507D-3 \\ -1.8177D-3 & 3.7549D-2 & -2.6245D-1 & -3.8672D-2 & -4.0230D-3 \\ -8.1458D-5 & 9.6849D-2 & 2.6870D-2 & 1.5885D-2 & -1.7085D-1 \end{bmatrix}$$

Table 5.15. Discrete PI Controller Gain Matrices - TFTA

$$K_1 = \begin{bmatrix} 1.3716D-1 & 8.3600D-1 & 3.8314D+0 & 8.7056D-2 & 8.9830D-1 \\ -1.5494D-1 & -8.3600D-1 & 3.9886D+0 & -8.7056D-2 & -8.9830D-1 \\ 2.7067D-1 & -4.6565D-1 & 2.3938D+0 & 6.9138D-1 & 9.3202D-3 \\ 2.6077D-1 & 4.6565D-1 & 2.3062D+0 & -6.9138D-1 & -9.3202D-3 \\ -1.5477D-2 & 1.4562D+0 & -1.3688D-1 & 1.9120D-1 & -1.1936D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 6.8582D-4 & 4.1800D-3 & 1.9157D-2 & 4.3528D-4 & 4.4915D-3 \\ 7.7468D-4 & -4.1800D-3 & 1.9943D-2 & -4.3528D-4 & -4.4915D-3 \\ 1.3533D-3 & -2.3282D-3 & 1.1969D-2 & 3.4569D-3 & 4.6601D-5 \\ 1.3038D-3 & 2.3282D-3 & 1.1531D-2 & -3.4569D-3 & -4.6601D-5 \\ -7.7385D-5 & 7.2808D-3 & -6.8439D-4 & 9.5602D-4 & -5.9680D-3 \end{bmatrix}$$

Table 5.16. Discrete PI Controller Gain Matrices - TFTA30TL

$$K_1 = \begin{bmatrix} 1.2984D-1 & 9.5872D-1 & 3.8718D+0 & -8.4379D-2 & 1.0234D+0 \\ 1.2931D-1 & -8.5960D-1 & 3.8668D+0 & -6.1076D-2 & -1.0147D+0 \\ 4.0678D-1 & -7.2395D-1 & 3.8332D+0 & 1.0625D+0 & -6.3369D-2 \\ 2.3628D-1 & 6.1259D-1 & 2.2265D+0 & -8.9901D-1 & 5.3621D-2 \\ -2.6797D-4 & 1.5908D+0 & -2.5251D-3 & 1.1652D-2 & -1.0190D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 6.4922D-4 & 4.7936D-3 & 1.9359D-2 & -4.2190D-4 & 5.1168D-3 \\ 6.4654D-4 & -4.2980D-3 & 1.9334D-2 & -3.0538D-4 & -5.0734D-3 \\ 2.0339D-3 & -3.6198D-3 & 1.9166D-2 & 5.3123D-3 & -3.1685D-4 \\ 1.1814D-3 & 3.0629D-3 & 1.1133D-2 & -4.4951D-3 & 2.6810D-4 \\ -1.3398D-6 & 7.9542D-3 & -1.2625D-5 & 5.8258D-5 & -5.0951D-3 \end{bmatrix}$$

Table 5.17. Discrete PI Controller Gain Matrices - TFTA50CL

$$K_1 = \begin{bmatrix} 4.0684D-1 & 2.2164D+0 & 1.1123D+1 & -1.6540D-2 & 2.1300D+0 \\ 2.0452D-1 & -2.2202D-1 & 5.5914D+0 & 1.6568D-3 & -2.1336D-1 \\ 3.7702D-1 & 4.5764D-1 & 6.1683D+0 & 6.4023D-1 & 8.8416D-1 \\ 3.6996D-1 & 1.1783D+0 & 5.9754D+0 & -6.5243D-1 & 6.8799D-1 \\ 1.1003D-3 & 1.1698D+0 & 3.0081D-2 & 9.9266D-3 & -1.2784D+0 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 2.0342D-3 & 1.1082D-2 & 5.5614D-2 & -8.2698D-5 & 1.0650D-2 \\ 1.0226D-3 & -1.1101D-3 & 2.7957D-2 & 8.2840D-6 & -1.0668D-3 \\ 1.8851D-3 & 2.2882D-3 & 3.0842D-2 & 3.2011D-3 & 4.4208D-3 \\ 1.8498D-3 & 5.8913D-3 & 2.9877D-2 & -3.2622D-3 & 3.4399D-3 \\ 5.5014D-6 & 5.8490D-3 & 1.5040D-4 & 4.9633D-5 & -6.3918D-3 \end{bmatrix}$$

Table 5.18. Discrete Time System Design Parameters (Step-Response Matrix)

Flight Condition	σ_1	σ_2	σ_3	σ_4	σ_5	$\pi_1 - \pi_5$	m_{31}	m_{42}
ACMENTRY	0.10	0.0015	0.02	0.04	0.6	0.04	0.40	0.20
ACM30TL	0.10	0.0015	0.02	0.04	0.6	0.04	0.40	0.20
ACM50CL	0.10	0.0015	0.02	0.04	0.6	0.04	0.40	0.20
ACMEXIT	0.04	0.0240	0.02	0.04	0.6	0.04	0.20	0.20
TFTA	0.10	0.0015	0.02	0.04	0.6	0.04	0.40	0.20
TFTA30TL	0.10	0.0015	0.02	0.04	0.6	0.04	0.40	0.20
TFTA50CL	0.10	0.0015	0.02	0.04	0.6	0.04	0.40	0.20

Table 5.19. Discrete PI Controller Gain Matrices (Step-Response Matrix) - ACMENTRY

$$\bar{K}_1 = \begin{bmatrix} 5.4705D+0 & 3.1920D+1 & 2.9162D+2 & 1.0949D+1 & 3.1691D+2 \\ 5.4705D+0 & -3.1920D+1 & 2.9162D+2 & -1.0949D+1 & -3.1691D+2 \\ 7.9174D+0 & -1.0367D+1 & -3.6901D+1 & 1.1511D+2 & -5.7625D+1 \\ 7.9174D+0 & 1.0367D+1 & -3.6901D+1 & -1.1511D+2 & 5.7625D-1 \\ 0.00000+0 & 5.4361D+1 & 0.00000+0 & 4.8890D+1 & 1.0767D-2 \end{bmatrix}$$

$$\bar{K}_2 = \begin{bmatrix} 5.6301D-4 & 1.6810D+0 & 1.3864D+0 & -7.6547D-1 & 2.1383D+1 \\ 5.6301D-4 & -1.6810D+0 & 1.3864D+0 & 7.6547D-1 & -2.1383D+1 \\ 5.4982D-4 & 2.7851D-1 & 1.5854D+0 & 2.4882D-1 & -6.9378D+0 \\ 5.4982D-4 & -2.7851D-1 & 1.5854D+0 & -2.4882D-1 & 6.9378D+0 \\ 0.00000-0 & 4.8837D+0 & 0.00000-0 & -1.3028D+0 & 3.6074D+1 \end{bmatrix}$$

Table 5.20. Discrete PI Controller Gain Matrices (Step-Response Matrix) - ACM30TL

$$\bar{K}_1 = \begin{bmatrix} 6.6586D+0 & -3.3663D+1 & 2.7674D-2 & 6.6423D+0 & 9.0754D-1 \\ 6.8479D+0 & 3.3862D+1 & 2.8193D-2 & -2.3309D+1 & -8.4704D-1 \\ 1.6887D+1 & -2.3336D+0 & -9.1662D+1 & 1.9572D-2 & -7.1049D-1 \\ 9.8894D+0 & 1.6797D-0 & -5.2221D-1 & -1.4088D-2 & 5.1140D-1 \\ 0.00000+0 & 6.0066D+1 & 0.0000D+0 & -7.2307D+1 & 3.0098D-2 \end{bmatrix}$$

$$\bar{K}_2 = \begin{bmatrix} 6.3001D-4 & -3.7384D+0 & 3.0899D+0 & 8.0528D-1 & -2.2195D-1 \\ 6.4661D-4 & 3.4894D+0 & 3.1724D+0 & -8.1238D-1 & 2.2402D+1 \\ 1.2083D-3 & 1.9510D+0 & 6.2449D+0 & 5.5666D-2 & -1.6248D-0 \\ 7.0833D-4 & -1.4485D+0 & 3.6601D+0 & -4.1328D-2 & 1.2063D-0 \\ 0.00000+0 & 3.9600D+0 & 0.00000+0 & -1.4400D+0 & 4.0041D+1 \end{bmatrix}$$

Table 5.21. Discrete PI Controller Gain Matrices (Step-Response Matrix) - ACM50CL

$$K_1 = \begin{bmatrix} -3.0678D+2 & -2.9452D+2 & -1.6858D+4 & 4.7280D+2 & -3.5658D+3 \\ -6.8803D+1 & -9.8568D+1 & -3.7809D+3 & 1.5824D+2 & -1.1934D+3 \\ -1.0118D+2 & -1.1887D+2 & -6.0192D+3 & 3.0917D+2 & -1.3947D+3 \\ -1.2709D+2 & -1.3368D+2 & -7.4428D+3 & 9.6259D+1 & -1.6629D+3 \\ -6.8803D+1 & -3.8514D+1 & -3.7809D+3 & 9.5510D+1 & -8.9256D+2 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -5.7289D-3 & -4.1622D-1 & -1.8685D+1 & 4.6308D-1 & -1.2974D+1 \\ -1.2848D-3 & -7.9454D-1 & -4.1906D+0 & 8.8399D-1 & -2.4767D+1 \\ -1.6573D-3 & 1.3764D-1 & -5.3147D+0 & 6.0436D-1 & -1.6920D+1 \\ -2.1411D-3 & -1.0420D+0 & -6.8925D+0 & 4.0179D-1 & -1.1270D+1 \\ -1.2848D-3 & 2.4855D+0 & -4.1906D+0 & -5.5601D-1 & 1.5254D+1 \end{bmatrix}$$

Table 5.22. Discrete PI Controller Gain Matrices (Step-Response Matrix) - ACMEXIT

$$K_1 = \begin{bmatrix} -2.3306D+0 & 3.4853D+2 & 1.5051D+3 & -6.7933D+0 & 4.6680D+2 \\ -2.2102D+0 & -3.5573D+2 & 1.4275D+3 & 7.4068D+0 & -4.7644D+2 \\ -6.5115D+0 & -2.7744D+2 & -1.3319D+3 & 1.0568D+3 & -6.1907D-1 \\ -5.8784D+0 & 2.8810D-2 & -1.7410D+3 & -1.0558D+3 & 7.6177D-1 \\ -2.6397D-1 & 7.4414D+2 & 1.7024D+2 & -2.3176D-2 & -9.0963D-2 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -1.6058D-1 & 1.0821D+0 & -1.8853D+1 & 1.8345D+0 & 1.7011D-1 \\ -1.5228D-1 & -9.9808D-1 & -1.6529D+1 & 2.6597D+0 & -1.5691D+1 \\ -3.3080D-1 & 1.8514D+0 & -3.6653D+1 & 5.4922D+0 & -1.1830D+1 \\ -2.8719D-1 & -1.6223D+0 & -3.2831D+1 & 4.1074D+0 & 1.5431D+1 \\ -1.8187D-2 & 1.4151D+0 & -3.5205D+0 & -7.3723D-1 & 3.4962D+1 \end{bmatrix}$$

Table 5.23. Discrete PI Controller Gain Matrices (Step-Response Matrix) - TFTA

$$K_1 = \begin{bmatrix} 2.7661D+0 & 1.0050D+1 & 2.1612D+2 & 3.6677D+0 & 9.4263D-1 \\ 3.1223D+0 & -1.0050D+1 & 2.2642D+2 & -3.6677D+0 & -9.4263D-1 \\ 5.4248D+0 & -5.5876D+0 & 1.5684D+2 & 6.4045D-1 & -2.7756D+1 \\ 5.2264D+0 & 5.5876D+0 & 1.5111D+2 & -6.4045D+1 & 2.7756D-1 \\ -3.1020D-1 & 1.7541D+1 & -8.9685D+0 & 9.6502D+0 & 2.9729D+1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1.3618D-3 & 1.2573D+0 & 1.1898D+0 & -2.1402D-1 & 6.7771D+0 \\ 1.5011D-3 & -1.2573D+0 & 1.3141D+0 & 2.1402D-1 & -6.7771D+0 \\ 2.1218D-3 & 2.8001D-1 & 1.8942D+0 & 1.1921D-1 & -3.7500D+0 \\ 2.0442D-3 & -2.8001D-1 & 1.8250D+0 & -1.1921D-1 & 3.7500D+0 \\ -1.2132D-4 & 4.0880D+0 & -1.0831D-1 & -3.7277D-1 & 1.1523D+1 \end{bmatrix}$$

Table 5.24. Discrete PI Controller Gain Matrices (Step-Response Matrix) - TFTA30TL

$$K_1 = \begin{bmatrix} 2.5976D+0 & 1.1521D+1 & 2.0106D+2 & -1.2369D+1 & 1.0779D-2 \\ 2.5870D+0 & -1.0332D+1 & 2.0080D+2 & -1.1203D+0 & -1.0136D+2 \\ 8.0892D+0 & -8.6979D+0 & 1.9769D+2 & 9.8675D+1 & -4.7011D-1 \\ 4.6987D+0 & 7.3430D+0 & 1.1483D+2 & -8.3304D+1 & 3.9688D-1 \\ -5.3287D-3 & 1.9154D+1 & -1.3022D-1 & -7.2938D+0 & 4.6405D-1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 9.4845D-4 & 1.0667D+0 & 9.1951D-1 & -2.4676D-1 & 7.8198D+0 \\ 9.4532D-4 & -1.1396D+0 & 9.1646D-1 & 2.1873D-1 & -6.9370D+0 \\ 2.3749D-3 & 4.9465D-1 & 2.3132D+0 & 1.9014D-1 & -5.9894D+0 \\ 1.3795D-3 & -4.0069D-1 & 1.3437D+0 & -1.5403D-1 & 4.8517D+0 \\ -1.5644D-6 & 3.8168D+0 & -1.5238D-3 & -4.0726D-1 & 1.2627D+1 \end{bmatrix}$$

5.3 Simulation and Results

The input commands to the plant for V, β, θ, ϕ, and r are applied for both a ramped step command and, a model following input based upon a third order response. In his thesis, Barfield provided a guideline to calculate the ramp time as a function of rate and position limit of the control surface [2].

$$Ramp\ Time = \frac{Position\ Limit}{Rate\ Limit} \tag{5.1}$$

The maximum rate and position limit for this design is calculated from the canard control surface as

$$\begin{aligned} Ramp\ Time &= \frac{60\ deg}{100\frac{deg}{sec}} && (5.2) \\ &= 0.6\ sec && (5.3) \end{aligned}$$

A ramp time of 0.5 second is used for all input channels to simplify MATRIX_x simulation. The ramped input is selected because the pilot cannot instantaneously provide a step input command when a maneuver is performed. Furthermore, first order prefilters are inserted to filter high frequency spikes in the input and provide a desirable bandwidth less than 10 rad/sec, thus eliminating the possibility of exciting structural modes in the aircraft [2]. For the ramped input, selecting a pre-filter of the form

$$Prefilter(s) = \frac{10}{s+10} \tag{5.4}$$

for all input channels reduces the bandwidth of the closed-loop system to approximately 8-10 rad/sec and does not noticeably affect the speed of the response.

The model following prefilters incorporate the desirable characteristic of reducing the bandwidth and are selected to provide a desired response, regardless of

Table 5.25. Discrete PI Controller Gain Matrices (Step-Response Matrix) - TFTA50CL

$$
\tilde{K}_1 = \begin{bmatrix} 8.0879D+0 & 2.6579D+1 & 5.7810D+2 & -9.1781D+0 & 2.3742D+2 \\ 4.0658D+0 & -2.7043D+0 & 2.9061D+2 & 9.3386D-1 & -2.4157D+1 \\ 7.4653D+0 & 5.4098D+0 & 3.1889D+2 & 5.5966D+1 & 6.9847D+1 \\ 7.3251D+0 & 1.4053D+1 & 3.0886D+2 & -6.2687D+1 & 1.0401D+2 \\ 2.1873D-2 & 1.4097D+1 & 1.5634D+0 & -3.4093D+0 & 8.2401D+0 \end{bmatrix}
$$

$$
\tilde{K}_2 = \begin{bmatrix} 2.9967D-3 & 3.3557D+0 & 2.5276D+0 & -5.5951D-1 & 1.7733D+1 \\ 1.5065D-3 & -3.6465D-1 & 1.2706D+0 & 6.0798D-2 & -1.9270D+0 \\ 2.4403D-3 & 1.5256D+0 & 2.0642D+0 & -8.5619D-2 & 2.7395D+0 \\ 2.3883D-3 & 6.0873D-1 & 2.0204D+0 & -2.7023D-1 & 8.5392D+0 \\ 8.1045D-6 & 3.5175D+0 & 6.8358D-3 & -2.9945D-1 & 9.2087D+0 \end{bmatrix}
$$

the input command. Settling time, overshoot, and bandwidth are determined from the values of the prefilter coefficients. This design is built and simulated from a transfer function having a one real root and a complex pair with a $\varsigma \approx 0.8$. The location of the real root affects the settling time and is adjusted to provide the desired response characteristics. The model following prefilter transfer functions and associated roots are listed in Table 5.26. The real roots (Table 5.26) are selected to meet anticipated pilot requirements; faster input tracking for velocity and bank angle commands and smoother and slower pitch angle tracking. The input commands for the coordinated turn and pitch tracking command for the ramped and model following input are shown in Figure 5.1 through 5.6.

Table 5.26. Model Following Transfer Functions

$$\frac{V_{input}}{V_{cmd}} = \frac{250}{s^3 + 18s^2 + 105s + 250}$$

$$\frac{\beta_{input}}{\beta_{cmd}} = \frac{100}{s^3 + 12s^2 + 57s + 100}$$

$$\frac{\theta_{input}}{\theta_{cmd}} = \frac{50}{s^3 + 10s^2 + 41s + 50}$$

$$\frac{\phi_{input}}{\phi_{cmd}} = \frac{250}{s^3 + 18s^2 + 105s + 250}$$

$$\frac{r_{input}}{r_{cmd}} = \frac{100}{s^3 + 12s^2 + 57s + 100}$$

Input	Denominator Roots	
$\frac{V_{input}}{V_{cmd}}$	-10 + 0j	-4 ± 3j
$\frac{\beta_{input}}{\beta_{cmd}}$	-4 + 0j	-4 ± 3j
$\frac{\theta_{input}}{\theta_{cmd}}$	-2 + 0j	-4 ± 3j
$\frac{\phi_{input}}{\phi_{cmd}}$	-10 + 0j	-4 ± 3j
$\frac{r_{input}}{r_{cmd}}$	-4 + 0j	-4 ± 3j

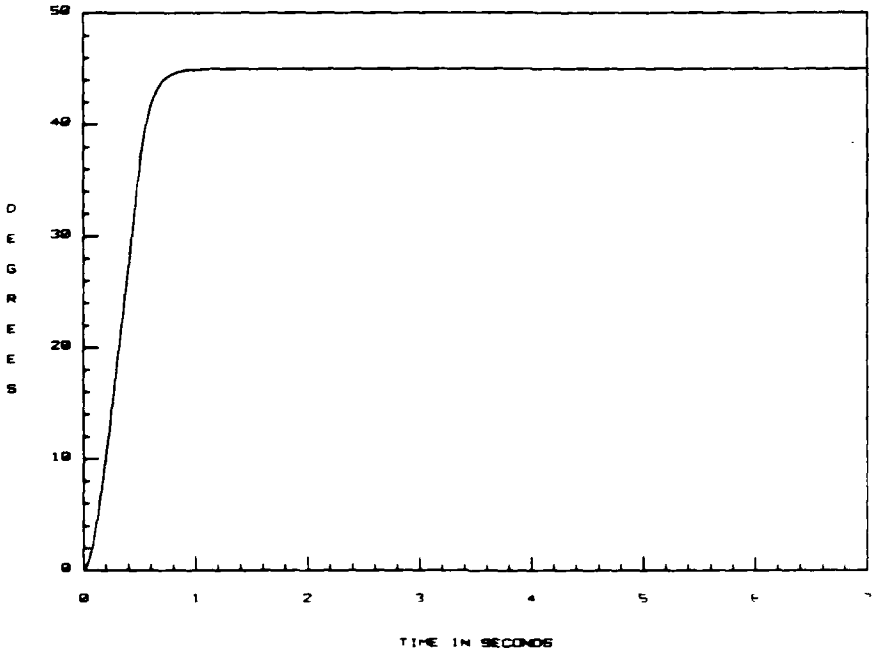


Figure 5.1. ϕ_{cmd}

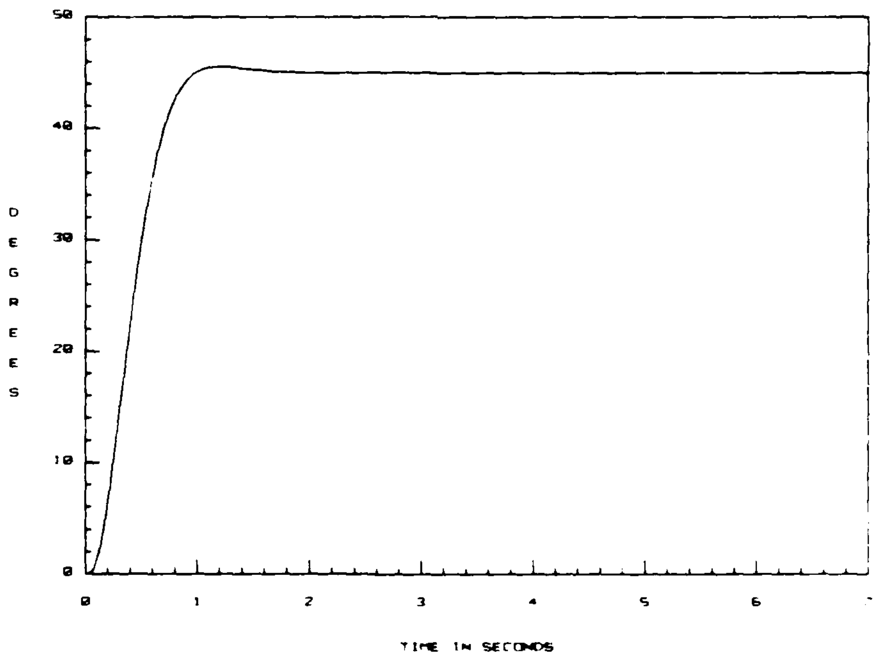


Figure 5.2. ϕ_{cmd} - Model Following

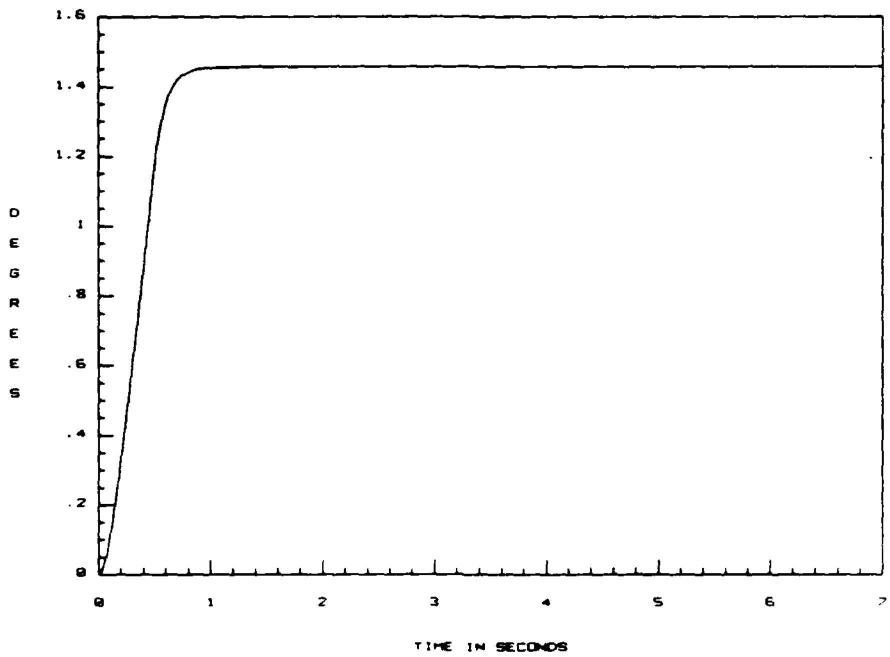


Figure 5.3. r_{cmd}

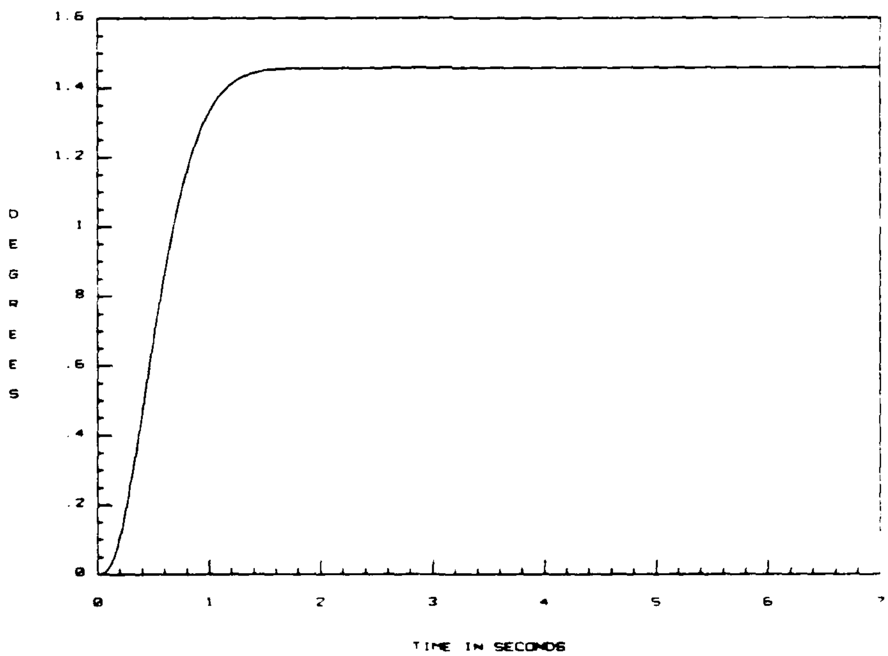


Figure 5.4. r_{cmd} - Model Following

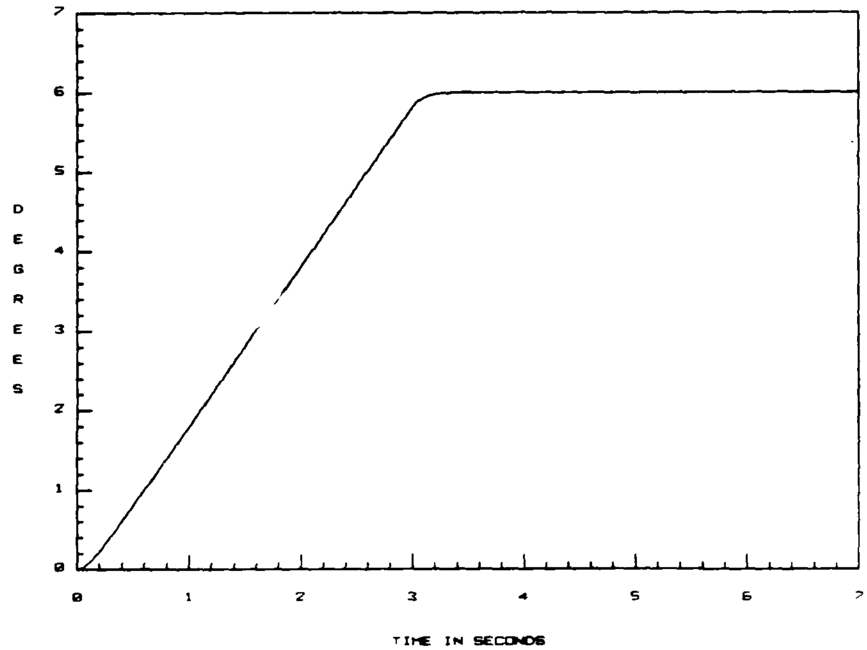


Figure 5.5. θ_{cmd}

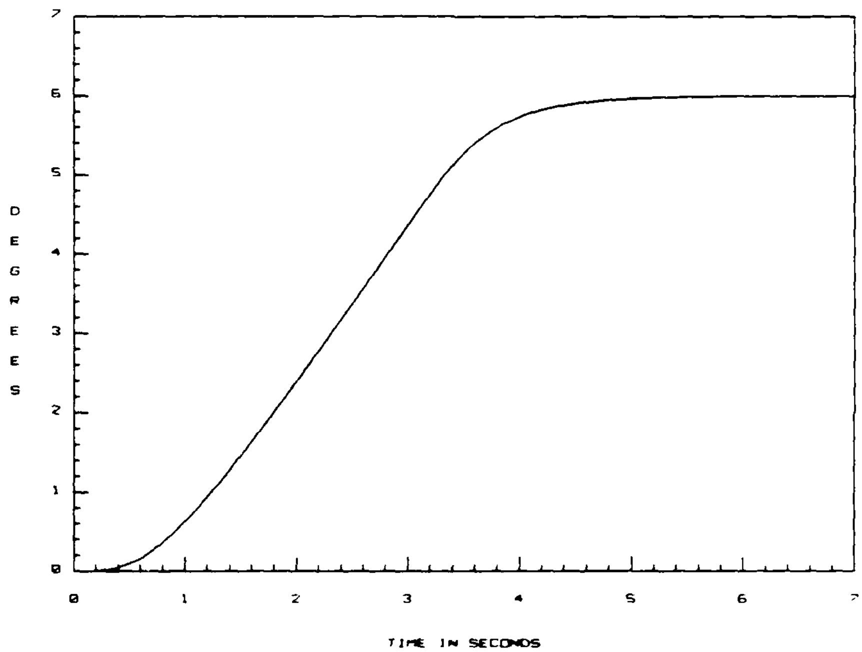


Figure 5.6. θ_{cmd} - Model Following

The $MATRIX_x$ simulation time is selected as seven seconds to provide adequate scaling during the first few seconds of the simulation. Figure 5.7 through 5.11 show the output responses for the continuous time PI controller for a ramped and model following input.

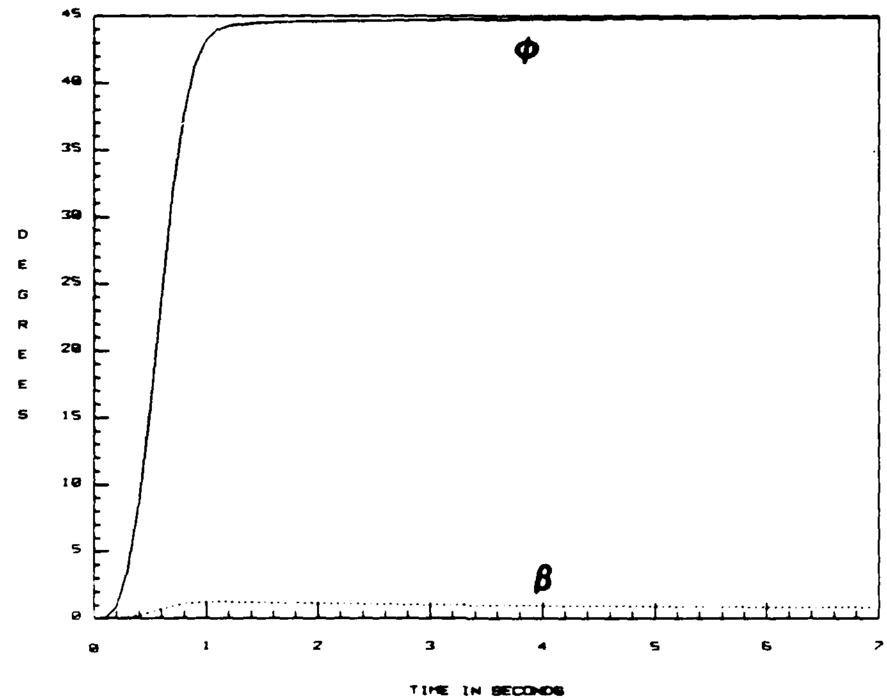


Figure 5.7. ϕ and β - 45° Banked Turn - (Continuous)

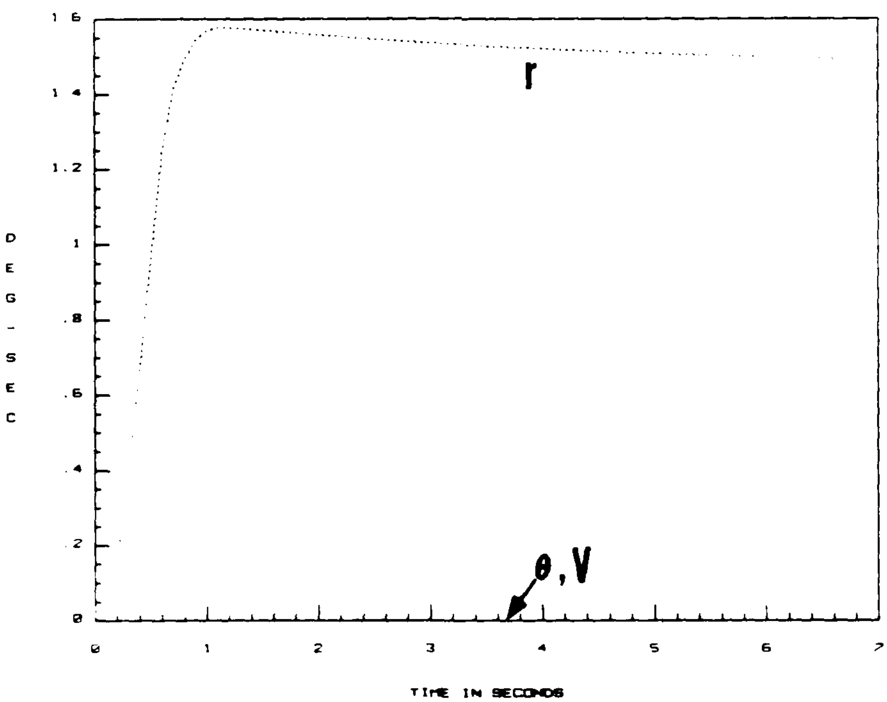


Figure 5.8. θ,r, and V - 45° Banked Turn - (Continuous)

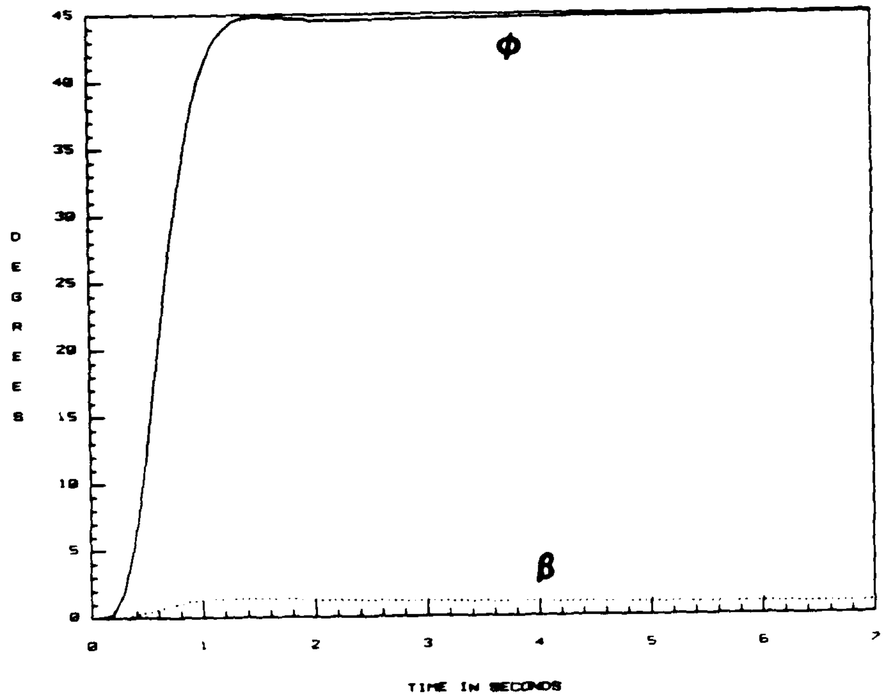


Figure 5.9. ϕ and β - 45° Banked Turn - Model Following (Continuous)

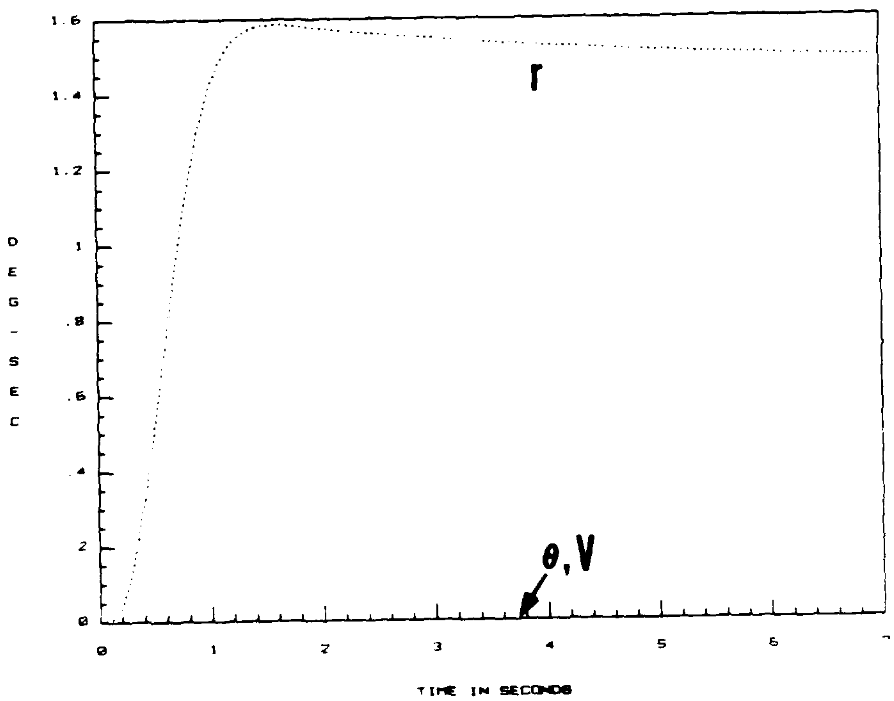


Figure 5.10. θ,r, and V - 45° Banked Turn - Model Following (Continuous)

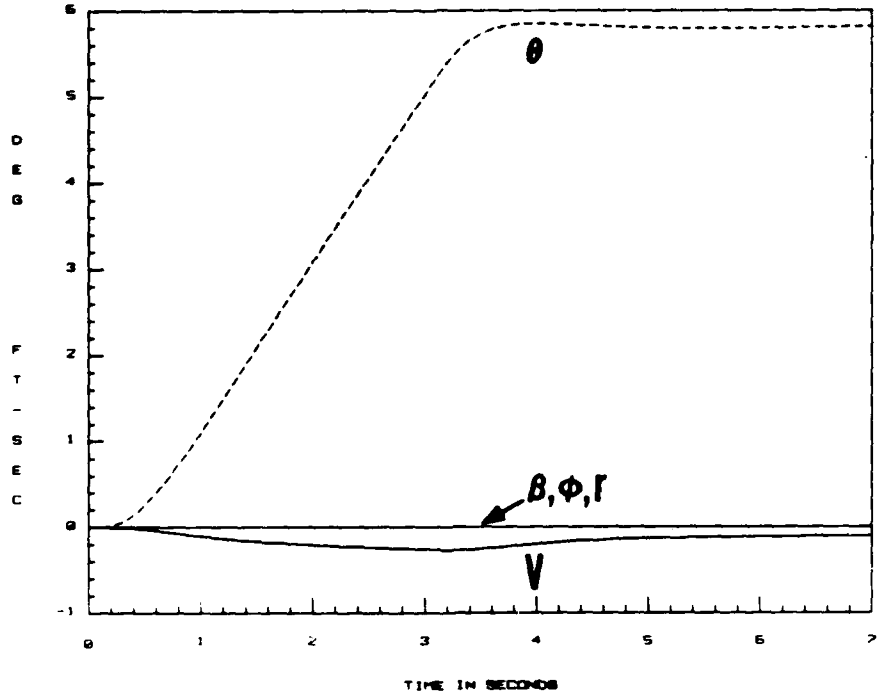


Figure 5.11. V,β,θ,ϕ, and r - θ_{cmd} - (Continuous)

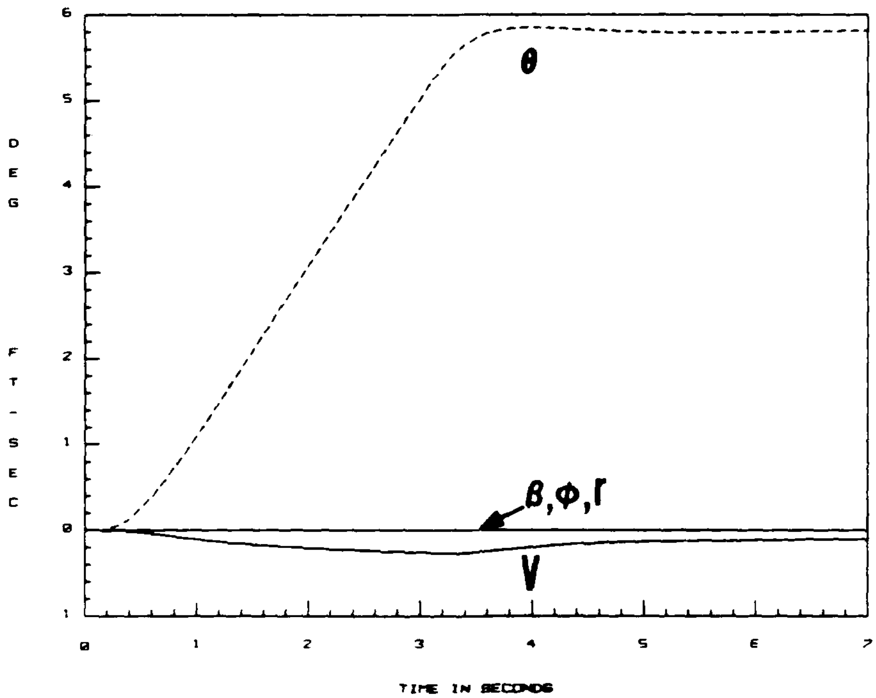


Figure 5.12. V,β,θ,ϕ, and r - θ_{cmd} - Model Following (Continuous)

The decoupling of the outputs is excellent for both the ramped and the model following input commands. There is a slight β output, less than one degree steady state, for the coordinated turn maneuver. If σ_2 of the Σ weighting matrix is increased to eliminate the output, excessive control surface deflections and rates occur. The one degree steady state error is an acceptable compromise between control surface deflection and desired decoupling. Decoupling is also present with the pitch tracking command with steady state value for the velocity output of less than 1.0 ft/sec. Appendix C illustrates the response for q, pitch rate, for the θ input command. The superb decoupling effect is present for all other flight conditions and failures. The responses for both the ramped and model following input look essentially the same. However, the control surface deflections and rates are much smalle. and smoother for the model following input, Appendix C.

The implementation of both types of discrete controllers produces similar decoupled outputs. A slight β interaction in the coordinated turn output and the velocity term, V, in the pitch tracking command is exhibited with the controllers. Figure 5.13 through 5.17 show the responses for the coordinated turn and pitch tracking tasks, respectively, using the basic discrete controller.

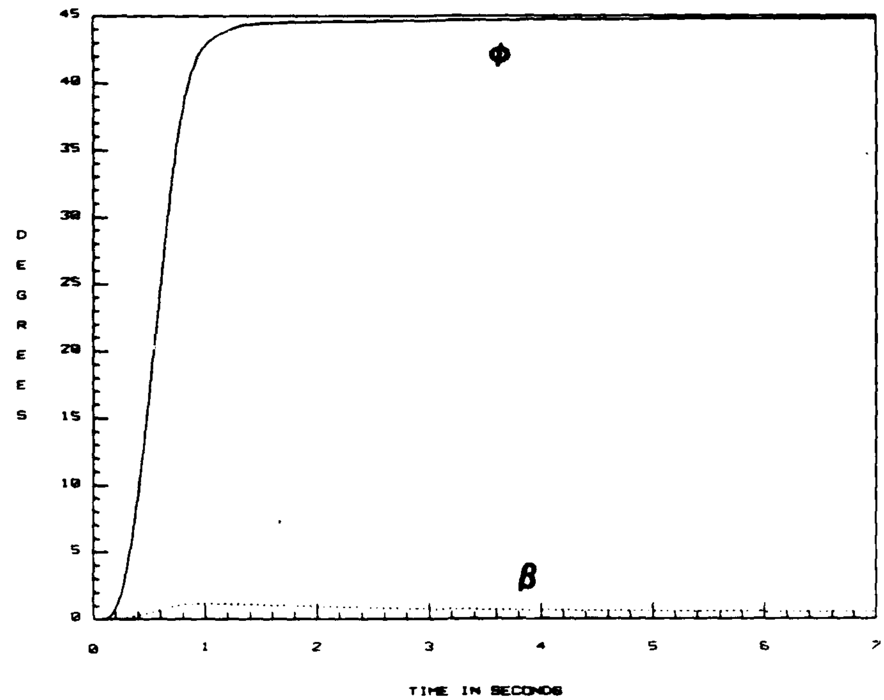


Figure 5.13. ϕ and β - 45° Banked Turn - (Discrete)

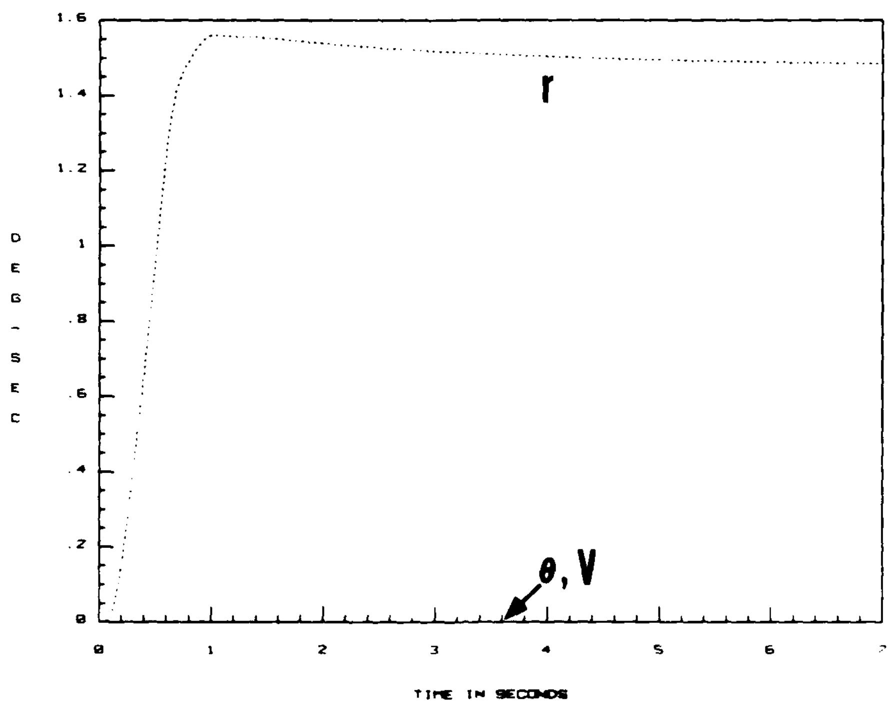


Figure 5.14. θ,r, and V - 45° Banked Turn - (Discrete)

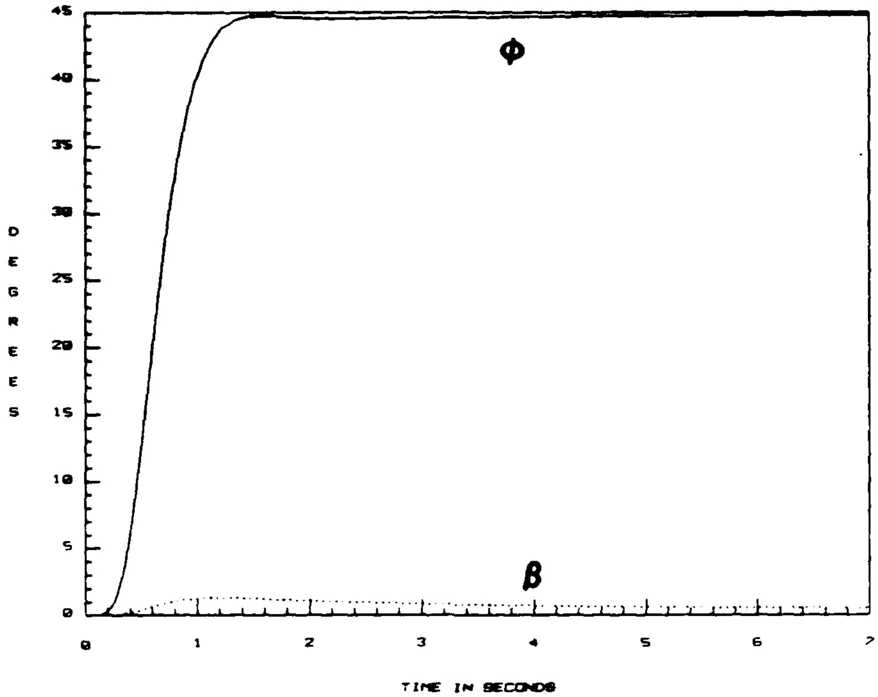


Figure 5.15. ϕ and β - 45° Banked Turn - Model Following (Discrete)

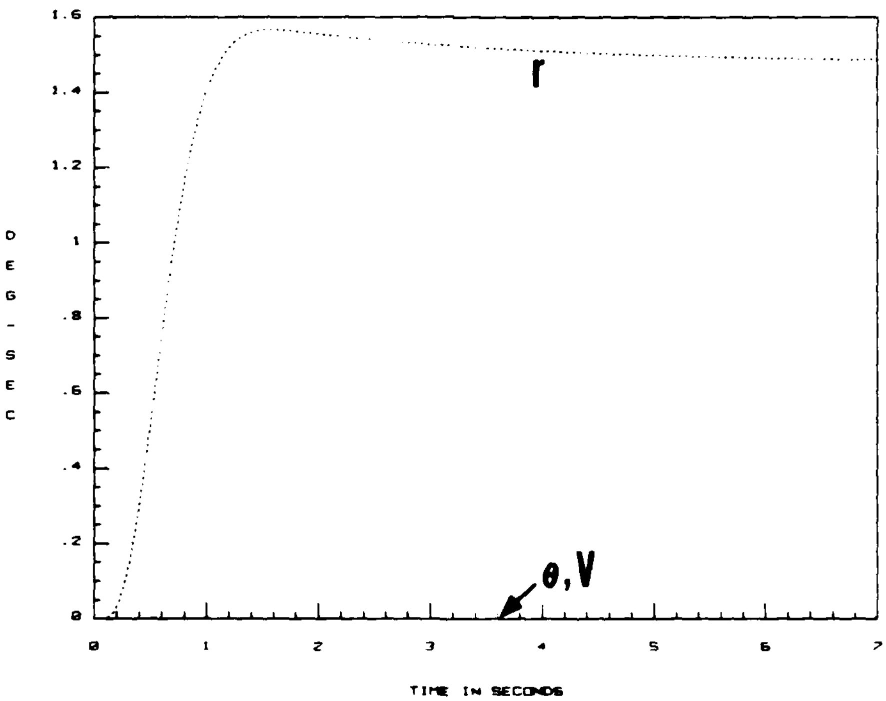


Figure 5.16. θ,r, and V - 45° Banked Turn - Model Following (Discrete)

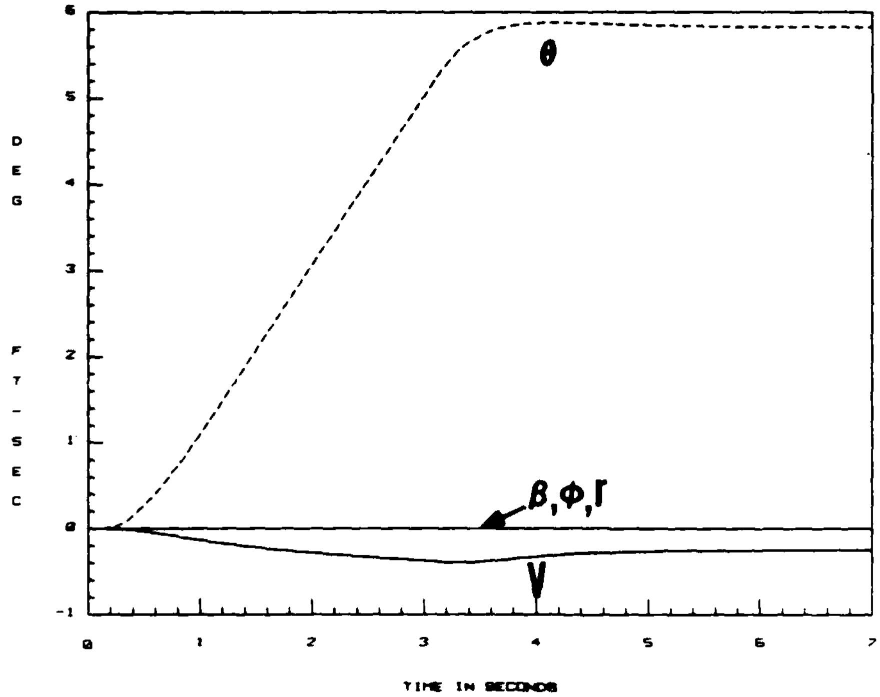


Figure 5.17. V,β,θ,ϕ, and r - θ_{cmd} - (Discrete)

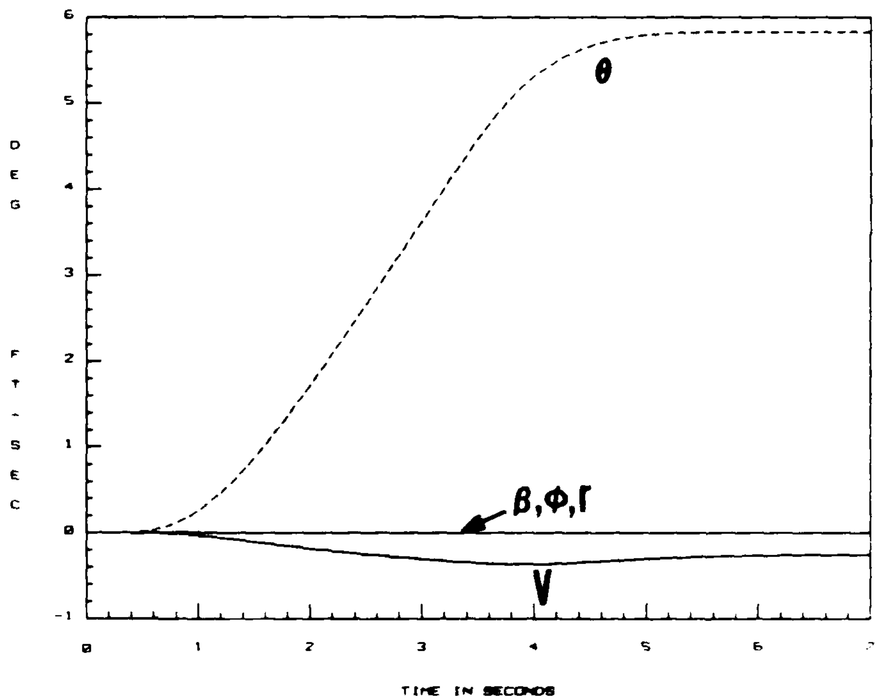


Figure 5.18. V,β,θ,ϕ, and r - θ_{cmd} - Model Following (Discrete)

The bandwidth of the closed loop system is calculated as the frequency, ω_b, where the response is down -3 dB. Figures 5.19 through 5.28 show the continuous closed loop frequency response for the ramped input with first order prefilter and the model following input. All responses indicate the bandwidth is less than 10 rad/sec. Similar results are evident for the discrete controller cases.

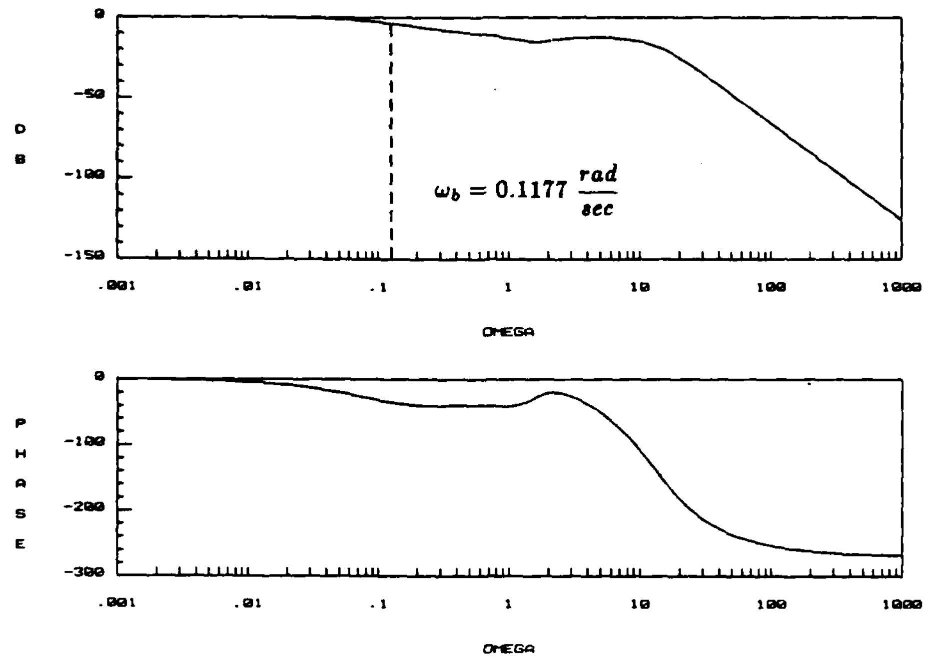


Figure 5.19. Continuous Closed-Loop Bode Plot - V vs V_{cmd}

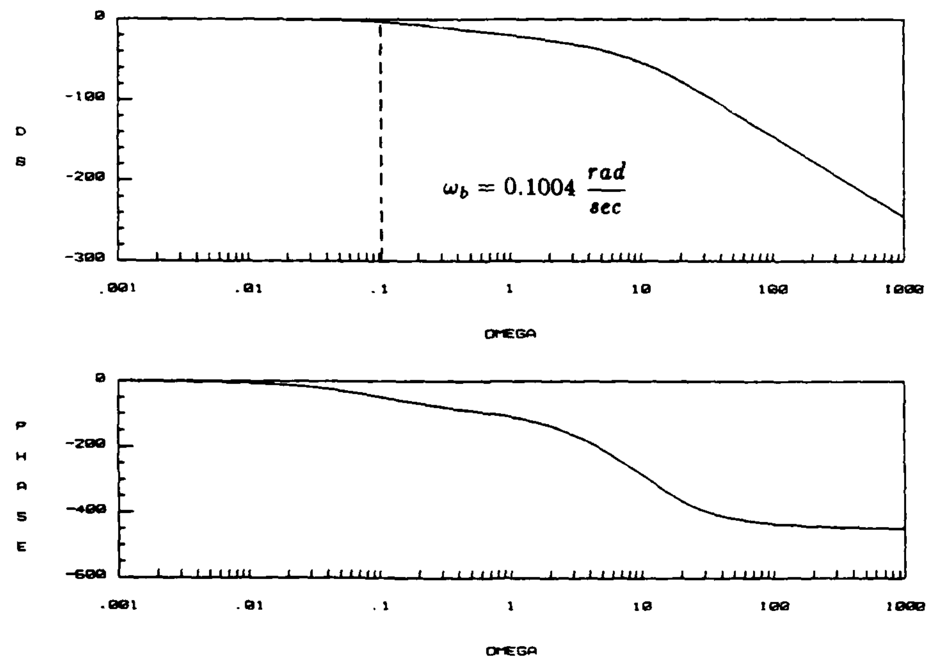


Figure 5.20. Continuous Closed-Loop Bode Plot - V vs V_{cmd} (Model Following)

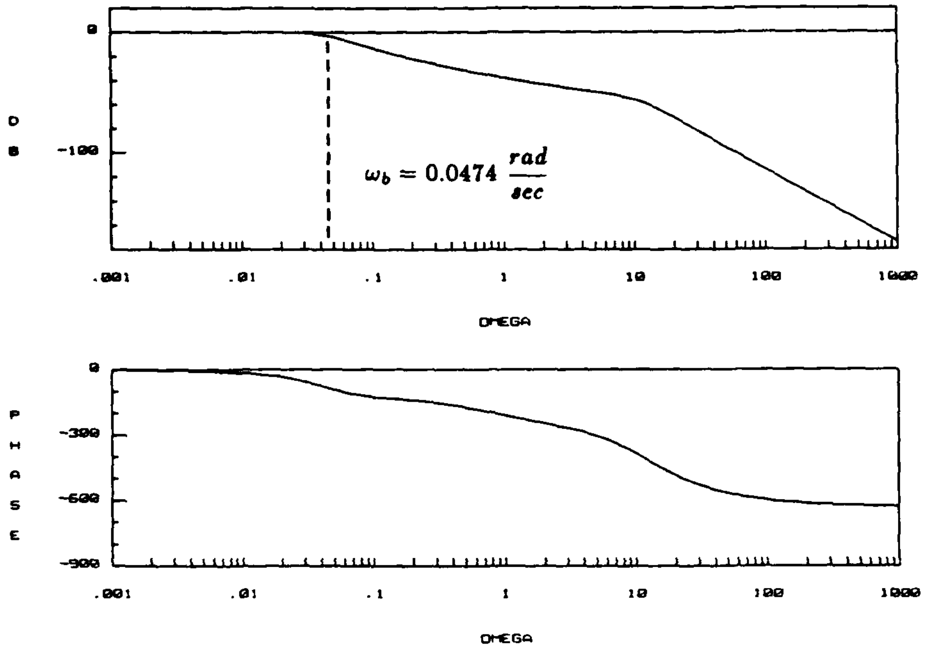


Figure 5.21. Continuous Closed-Loop Bode Plot - β vs β_{cmd}

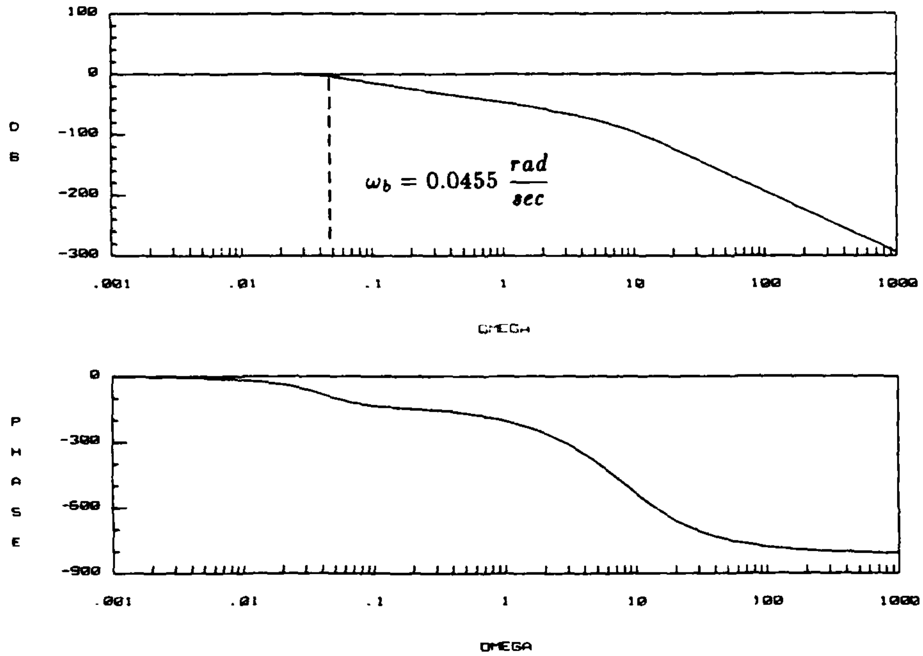


Figure 5.22. Continuous Closed-Loop Bode Plot - β vs β_{cmd} (Model Following)

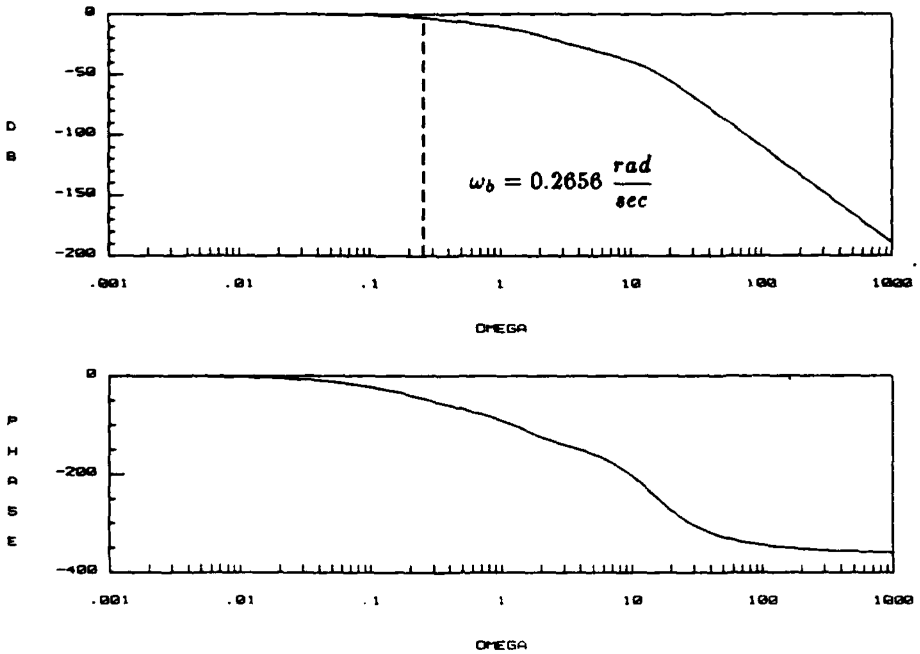


Figure 5.23. Continuous Closed-Loop Bode Plot θ vs θ_{cmd}

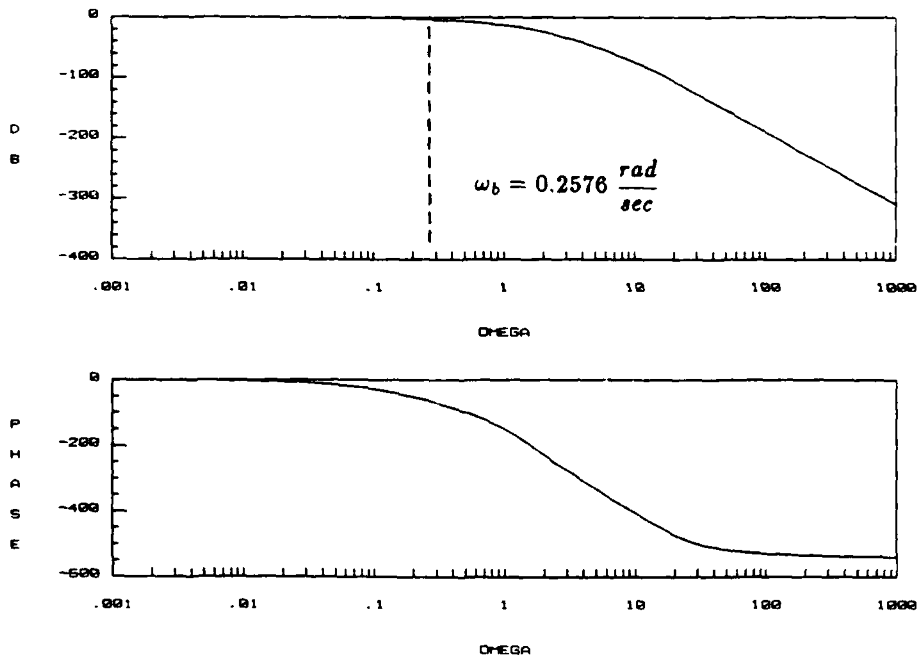


Figure 5.24. Continuous Closed-Loop Bode Plot θ vs θ_{cmd} (Model Following)

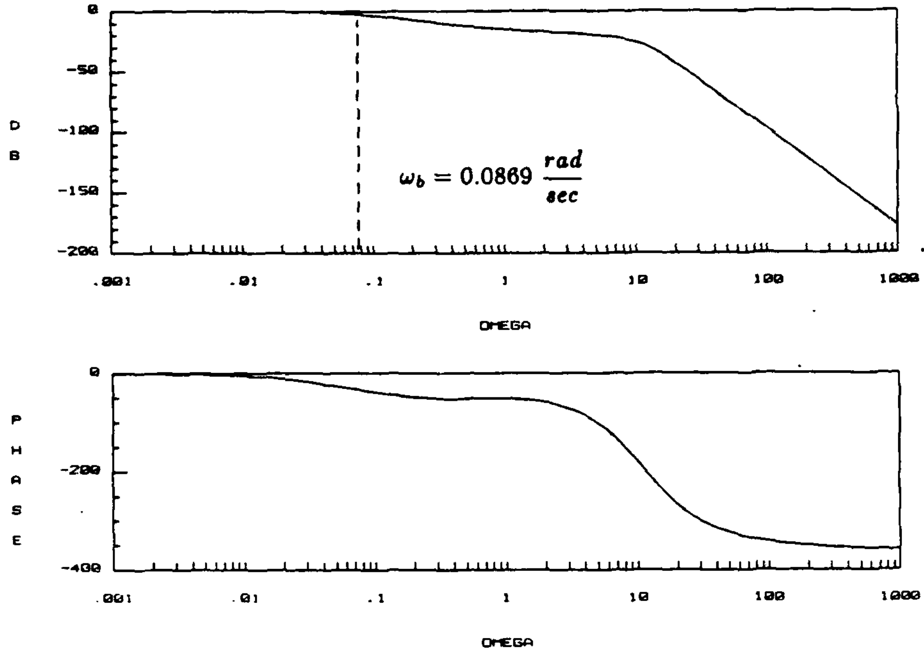


Figure 5.25. Continuous Closed-Loop Bode Plot ϕ vs ϕ_{cmd}

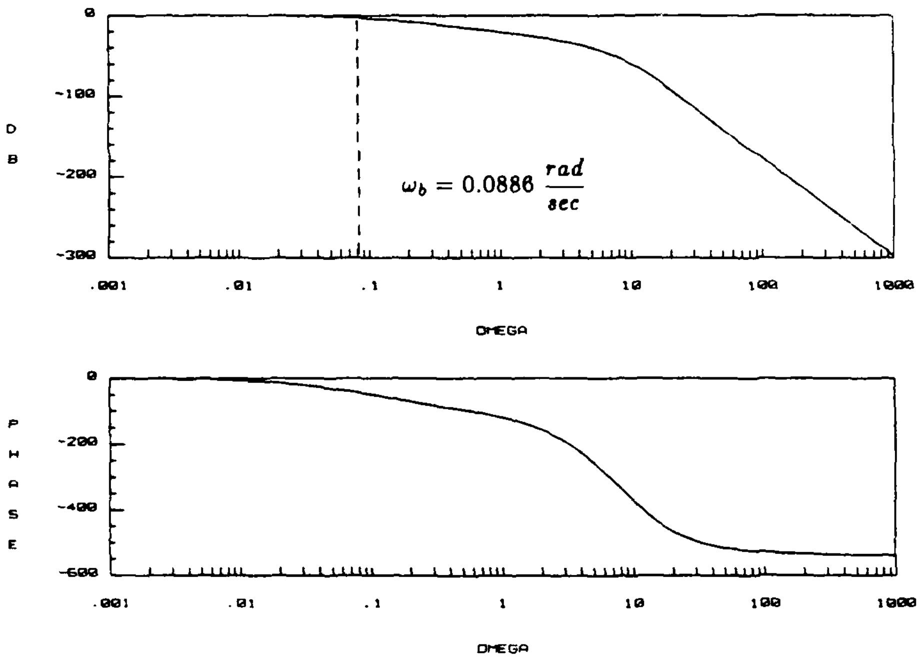


Figure 5.26. Continuous Closed-Loop Bode Plot ϕ vs ϕ_{cmd} (Model Following)

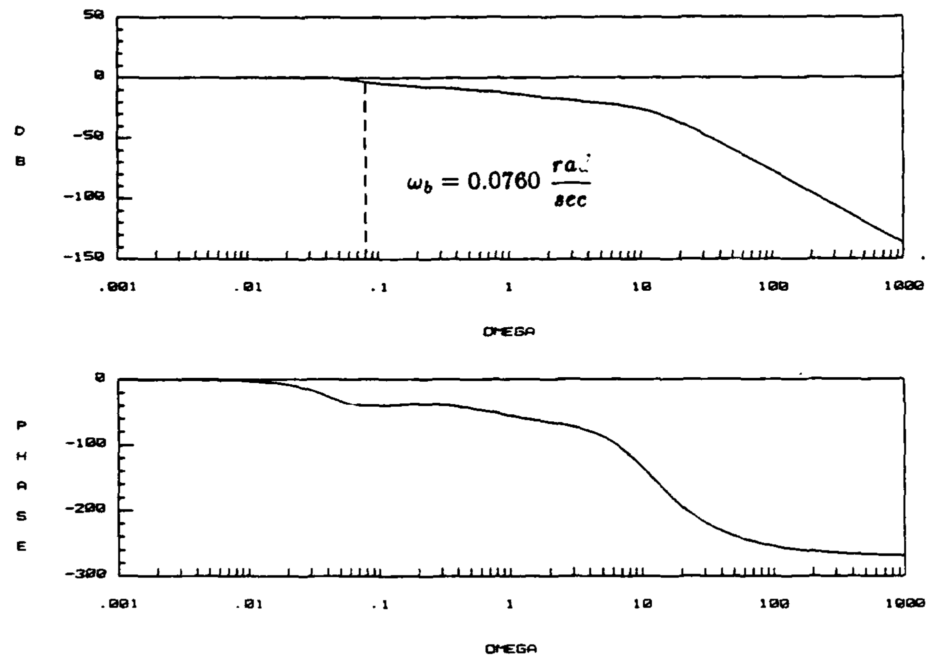


Figure 5.27. Continuous Closed-Loop Bode Plot - r vs r_{cmd}

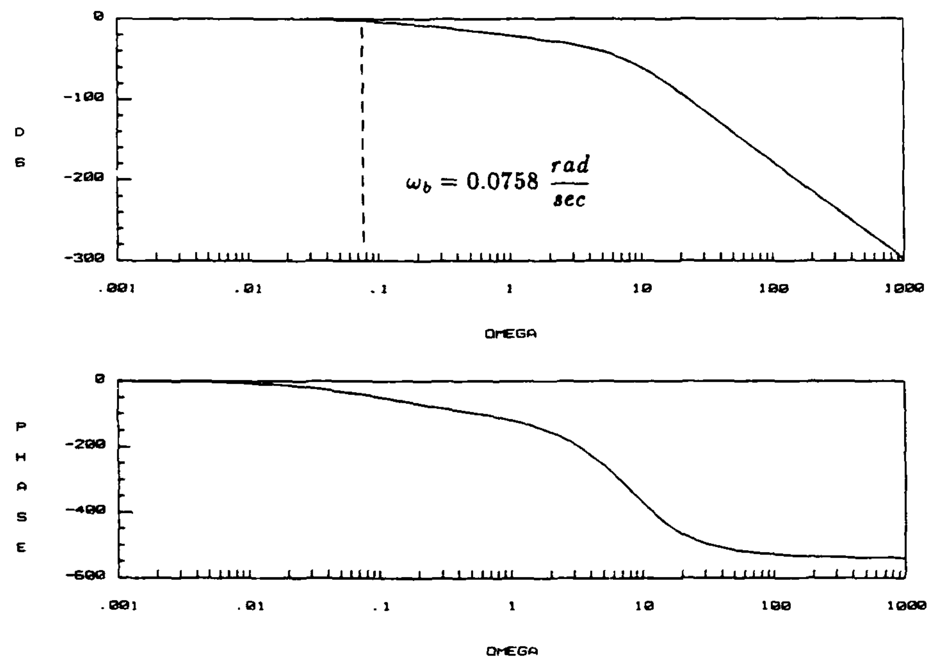


Figure 5.28. Continuous Closed-Loop Bode Plot - r vs r_{cmd} (Model Following)

5.4 Summary

One of the primary attributes of the PI control law developed by Professor Brian Porter is the ability to decouple the output responses which enhances the fast tracking capability of the controller. Optimal selection of the PI control law gains provides good tracking characteristics while maintaining control surface deflections and rates within acceptable limits. Ideally, the control engineer desires to select a set of gains that performs satisfactorily over a wide range of operating conditions. The thrust of the design effort in this research project was made to find a set of weighting elements of the Σ matrix that is acceptable for all conditions, thus simplifying gain scheduling efforts if a fixed gain controller is implemented. Often control law gains that are calculated on plant characteristics at high $\bar{q}$ conditions, such as the TF/TA flight segment, perform well at other flight conditions. In this research project, the gain matrices calculated at the TF/TA flight segment yielded stable responses at 70 percent of the remaining points chosen in the flight envelope. Adjustment of the controller gains was required to obtain good stable responses at the other flight conditions. An adaptive controller, such as the one presented in the next chapter, is needed to adjust the control law gains when universally selected gains do not work.

VI. Parameter Adaptive Controller Design

6.1 Introduction

As flight conditions and configurations change, the properties or parameters representing the aircraft plant also change. If the fixed gain PI controller does not provide a satisfactory response, the control law gains must be adjusted on-line to compensate for these changing conditions. The continuous and discrete controller gains developed in Chapters 4 and 5 are calculated using the state-space representation of the aircraft. Since the aircraft plant can be represented by a discrete difference equation for a given sampling time T, control law gains are calculated by manipulating the coefficients of this difference equation. By measuring input and output data, on-line, the coefficients that constitute the difference equation itself can be updated continuously and new PI controller gains calculated each time the measurements are taken. Predicting the new plant parameters allows gain calculations to be accurate and appropriate for the new flight condition or configuration. The recursive least squares (RLS) parameter estimating scheme is implemented in this thesis to provide updated step-response matrices for use in the PI control law gain calculations. The autoregressive moving average (ARMA) representation of the eight state CRCA plant for a 40 Hz sampling frequency produces a 2nd order discrete difference equation with 100 plant parameters. Using the reduceu order difference equation provided by the ARMA representation permits implementation of the RLS algorithm that estimates *all* plant parameters. This chapter describes the design procedure and results for the parameter- adaptive PI controller when a partially failed left trailing edge flaperon is introduced during the commanded 45 degree banked turn simulation. This failure is selected because gain scheduling is needed to provide a stable response at this flight condition, Table 5.1.

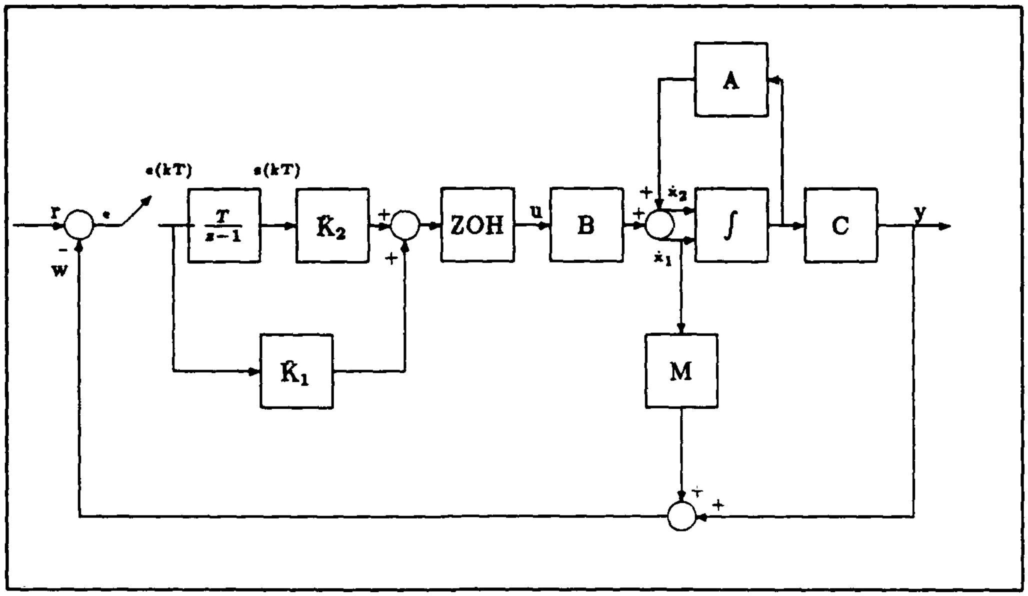


Figure 6.1. Discrete PI Controller - Step-Response Matrix

6.2 Design Procedure

The development of the fixed gain discrete PI controller in Chapter 4 provides a design basis for selection of the controller weighting matrices Π and Σ of Equations 3.31 and 3.32. The control law of Equation 3.30 and Figure 6.1 is shown as

$$u(kT) = \bar{K}_1 e(kT) + \bar{K}_2 z(kT)$$

where,

$$\begin{aligned}
\bar{K}_1 &= H(T)^T[H(T)H(T)^T]^{-1}\Sigma \\
\bar{K}_2 &= G(0)^T[G(0)G(0)^T]^{-1}\Pi \\
\Sigma &= \text{diagonal weighting matrix } [\sigma_1, \ldots \sigma_m] \\
\Pi &= \text{diagonal weighting matrix } [\pi_1, \ldots \pi_m]
\end{aligned}$$

Table 6.1. Control Law Design Parameters

σ_1	σ_2	σ_3	σ_4	σ_5	$\pi_1 - \pi_5$	m_{31}	m_{42}
0.1	0.0015	0.02	0.04	0.6	0.04	0.40	0.20

The selection of the design parameters for the control law are listed in Table 6.1 and represent those values obtained from fixed gain simulations that are listed in Table 4.14.

The behavior of the plant is modelled at each sampling period T by means of an autoregressive difference equation, Equation 2.8, of the form

$$\begin{aligned} y(kT) &= B_1 u[(k-1)T] - A_1 y[(k-1)T] + B_2 u[(k-2)T] - \\ &\quad A_2 y[(k-2)T] \end{aligned} \tag{6.1}$$

where,

A_1, A_2, B_1 and B_2 are the coefficients or parameters of the system and are matrices of dimension 5 x 5. Appendix B develops the method used to obtain the ARMA model representation used in this thesis and Appendix A contains the ARMA coefficients for all flight conditions. For the recursive least squares estimation, Equation 6.1 is expressed as

$$y(kT) = \theta^T \phi \tag{6.2}$$

where,

$$\theta^T = [A_1 \; A_2 \; B_1 \; B_2] \tag{6.3}$$

$$\phi^T = [-y^T[(k-1)T] \ldots - y^T[(k-2)T] \;\; u^T[(k-1)T \ldots u^T[(k-2)T] \tag{6.4}$$

The vector θ^T is of dimension 1 x 100 which is equal to the number of parameters in the difference equation, and ϕ^T is a matrix of the past values of the inputs and outputs of the system and has dimension 5 x 100.

The step-response matrix is presented in Chapter 3 in Equations 3.35 and 3.36 in terms of the difference equation coefficients. For this design, these equations are

$$H(T) = B_1 \tag{6.5}$$

$$G(0) = (I + A_1 + A_2)^{-1}(B_1 + B_2) \tag{6.6}$$

For simplicity and ease of design, the RLS algorithm used in this thesis does not use a variable forgetting factor. Consequently, the only design parameter in the algorithm is the α_t weighting term in Equation 3.41. Equations 3.39 through 3.42 are reproduced here to aid in the design discussion.

$$\theta(k+1) = \theta(k) + P(k)x(k+1)\gamma(k+1)[y(k+1) - x^T(k+1)\theta(k)]$$

$$P(k+1) = P(k) - P(k)x(k+1)\gamma(k+1)x^T(k+1)P(k)$$

$$\gamma(k+1) = \frac{1}{[\alpha_t + x^T(k+1)P(k)x(k+1)]}$$

$$x(k+1) = [-y(k), \ldots, -y(k-1), u(k), \ldots, u(k-1)]$$

where,

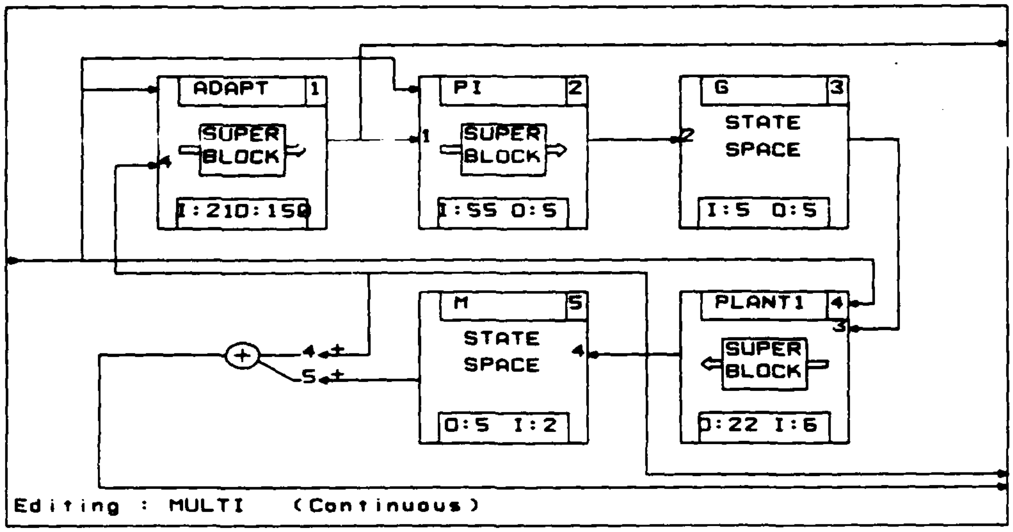


Figure 6.2. System Build Implementation of Adaptive Control Law

$\theta(k)$ = plant parameter vector (100 x 1)

$P(k)$ = covariance matrix associated with estimation (100 x 100)

$x(k-1)$ = time history of input/output data (100 x 5)

α_t = weighting parameter - diag(5 x 5)

The RLS algorithm is implemented by a FORTRAN subroutine and accessed each sample period during the MATRIX_x simulation. Figure 6.2 shows the MATRIX_x System Build implementation with the block labeled "ADAPT" containing the RLS FORTRAN program. Appendix F details the coding of the software.

The algorithm start-up values for the covariance matrix $P(0)$ and time history matrix $x(0)$ are selected as the identity matrix and a matrix of zeros respectively. However, to avoid any inversion singularities and to insure full population of the

Table 6.2. Adaptive Controller Design Parameters

α_t	σ_1	σ_2	σ_3	σ_4	σ_5	$\pi_1 - \pi_5$	m_{31}	m_{42}
0.94	0.1	0.0015	0.02	0.04	0.6	0.04	0.40	0.20

time history matrix, x(k+1), initial controller gains are kept constant for two sample periods, 0.050 second. These gain values are calculated from the nominal flight condition ARMA model coefficients. The rate of parameter adaption and overall stability of the output response is controlled by the selection of the weighting parameter α_t. The best output and rate of adaption occurs when this value is selected as 0.94. Table 6.2 summarizes all the design parameters used in the adaptive controller.

6.3 Simulation and Results

Responses for the 45 degree coordinated turn maneuver are shown starting with Figure 6.3 and illustrate the exceptional performance of the adaptive controller. The adaptive controller is functioning from the start of the simulation and the partially failed left trailing edge control surface is introduced at two seconds. The aircraft remains stable during the transient period of adaption with minimal bank angle fluctuations. There is some oscillation in the yaw rate value, but tracking convergence on the yaw rate command is obtained within three seconds. Control surface deflections and rates are within their limits.

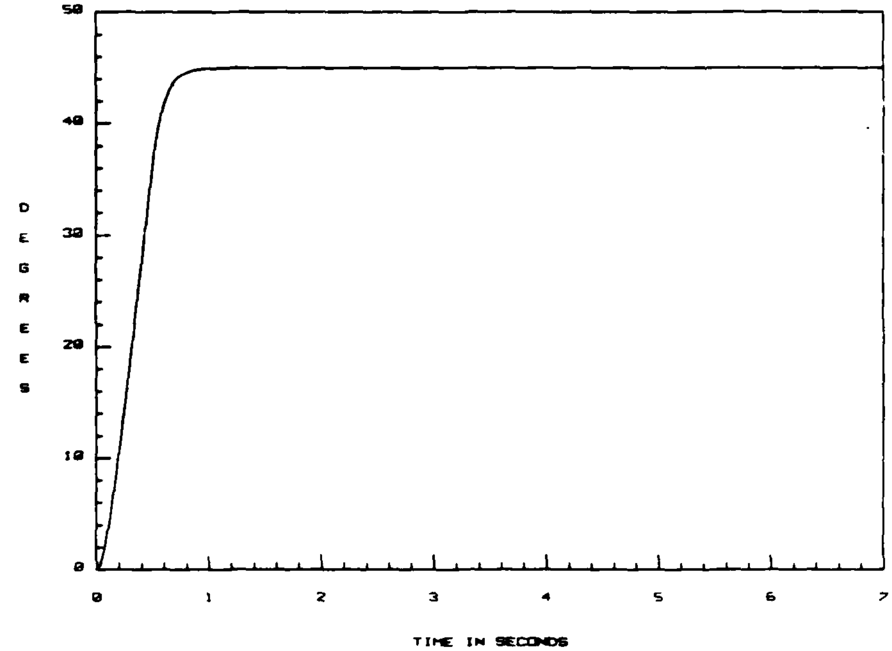


Figure 6.3. ϕ_{cmd} - 45° Banked Turn

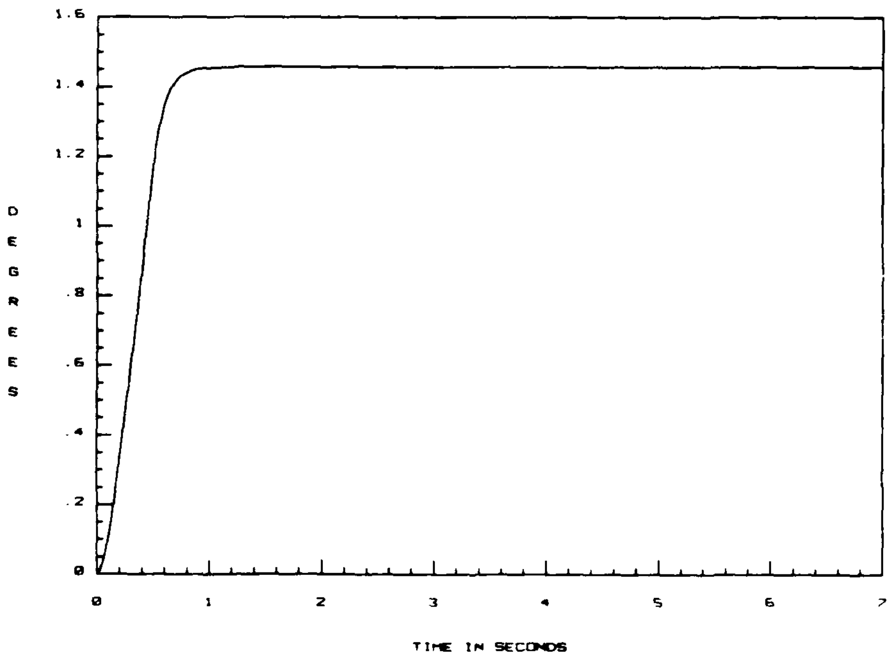
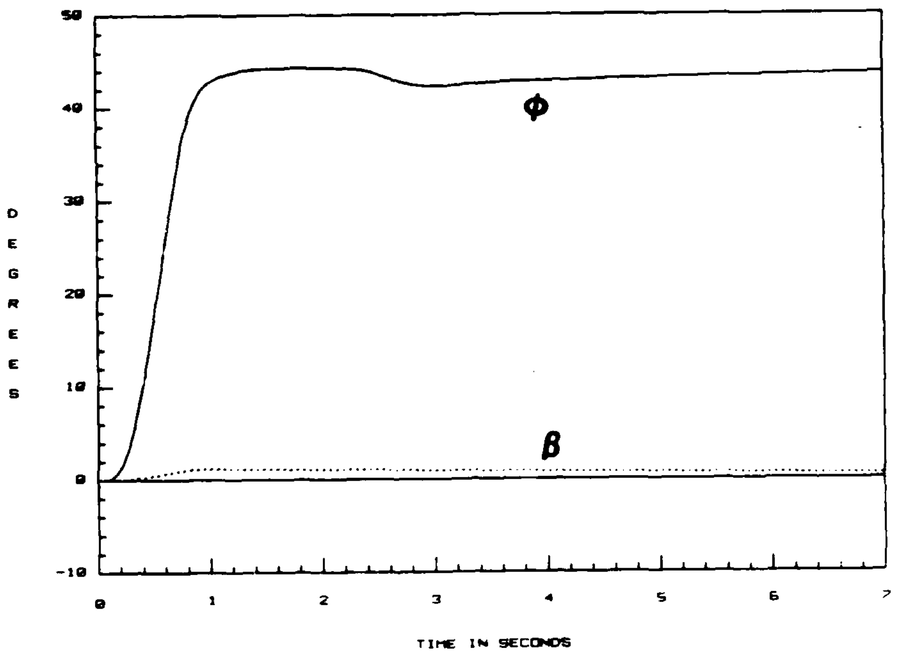


Figure 6.4. r_{cmd} - 45° Banked Turn



Figu-e 6.5. ϕ, β - 45° Banked Turn

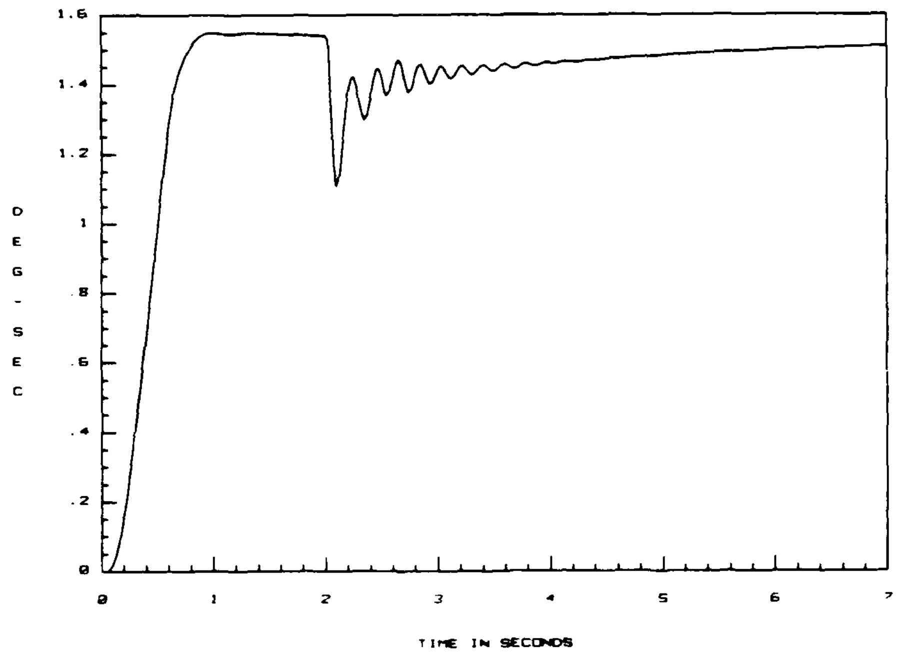


Figure 6.6. r - 45° Banked Turn

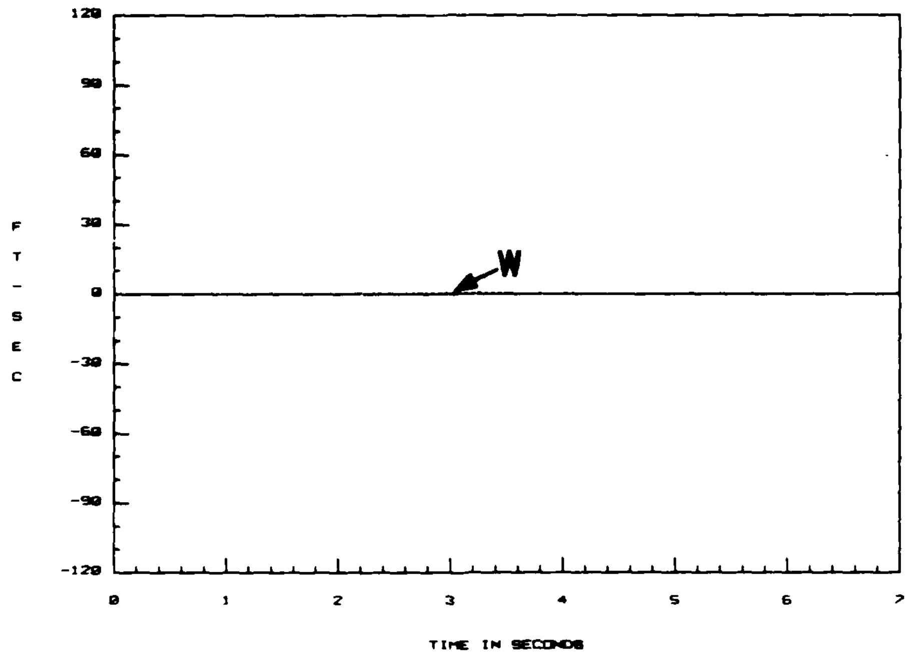


Figure 6.7. u and w - 45° Banked Turn

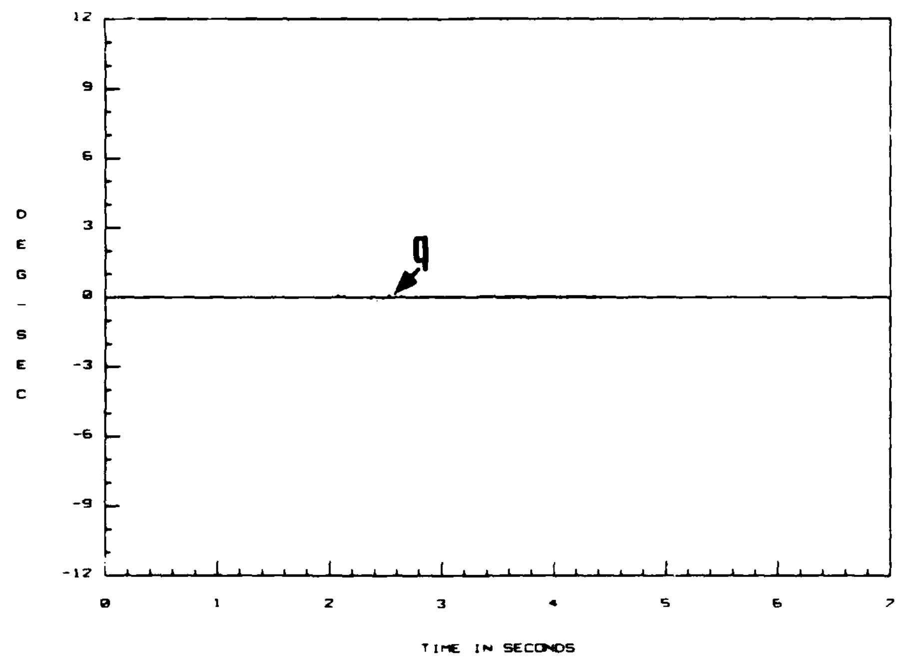


Figure 6.8. θ and q - 45° Banked Turn

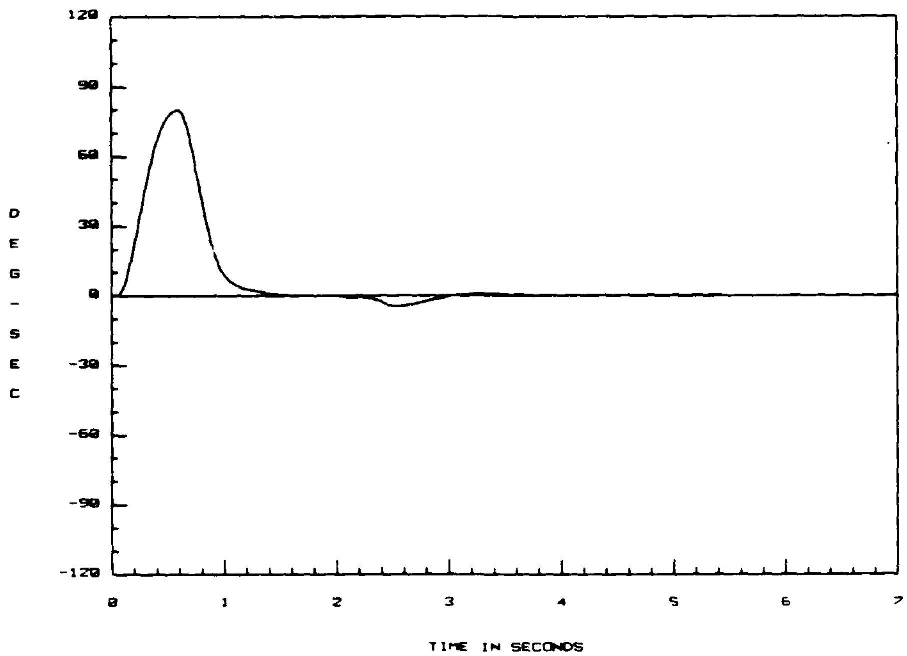


Figure 6.9. p - 45° Banked Turn

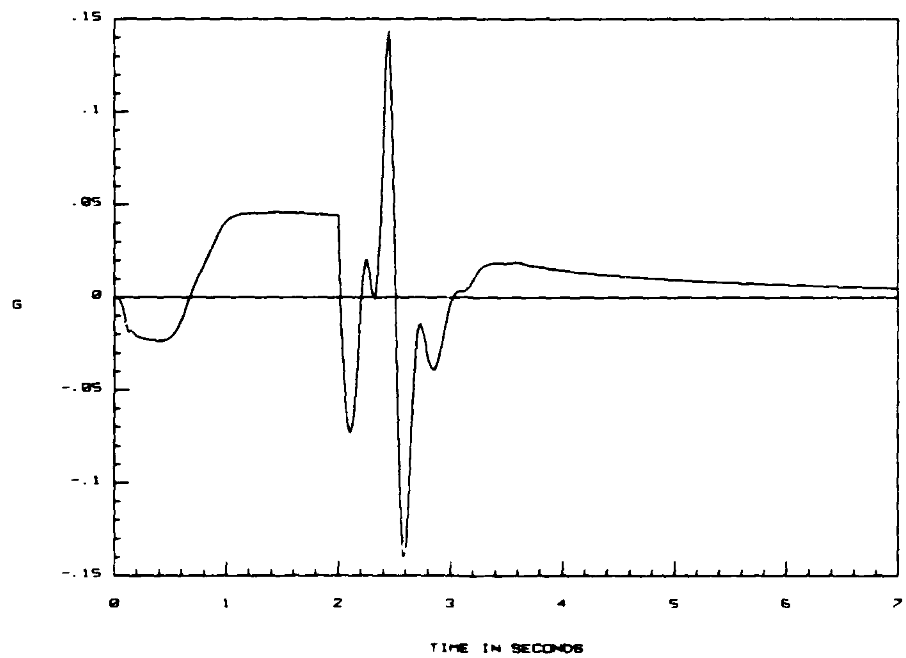


Figure 6.10. Normal Acceleration - 45° Banked Turn

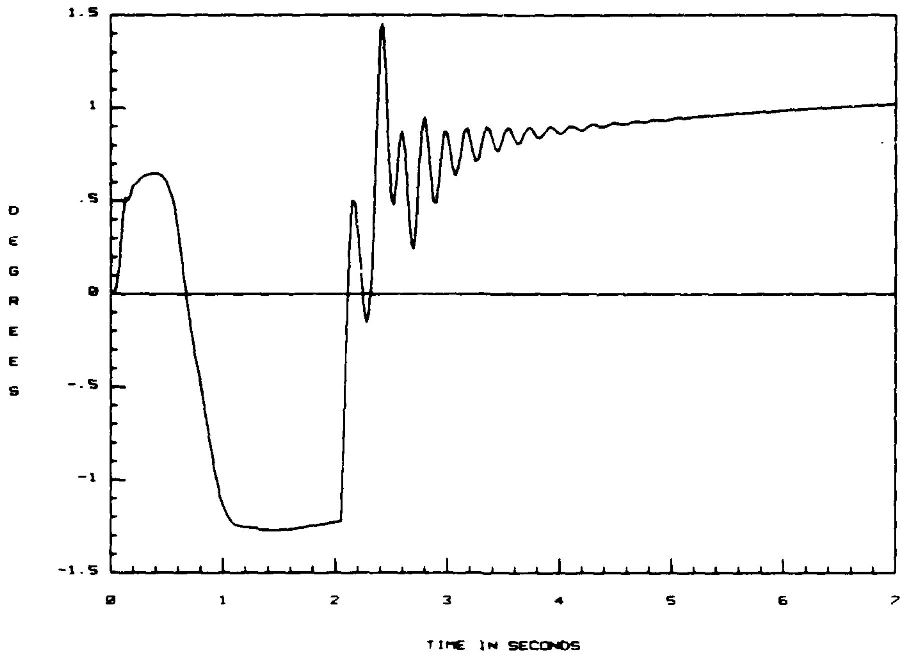


Figure 6.11. Left Canard Deflection - 45° Banked Turn

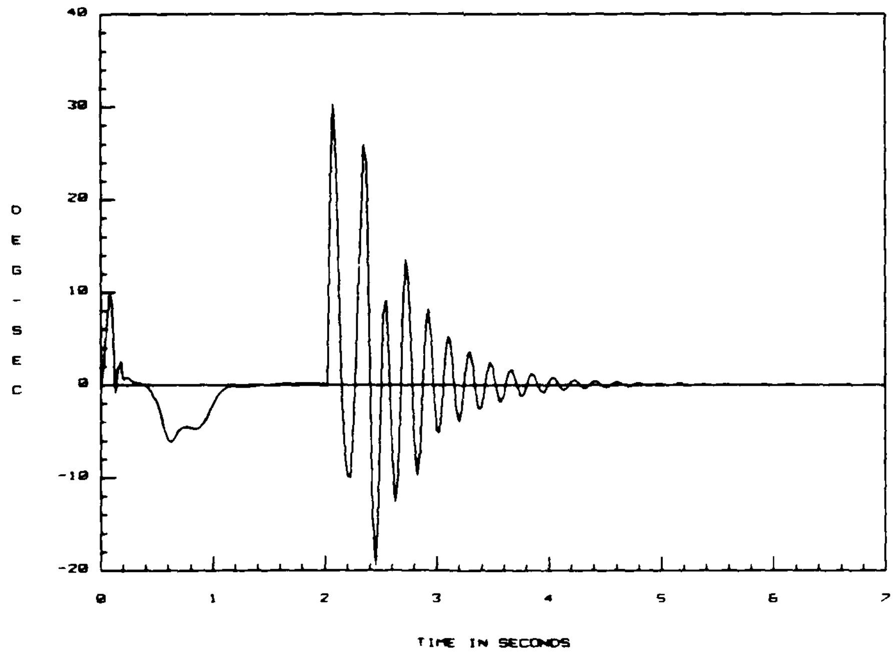


Figure 6.12. Left Canard Deflection Rate - 45° Banked Turn

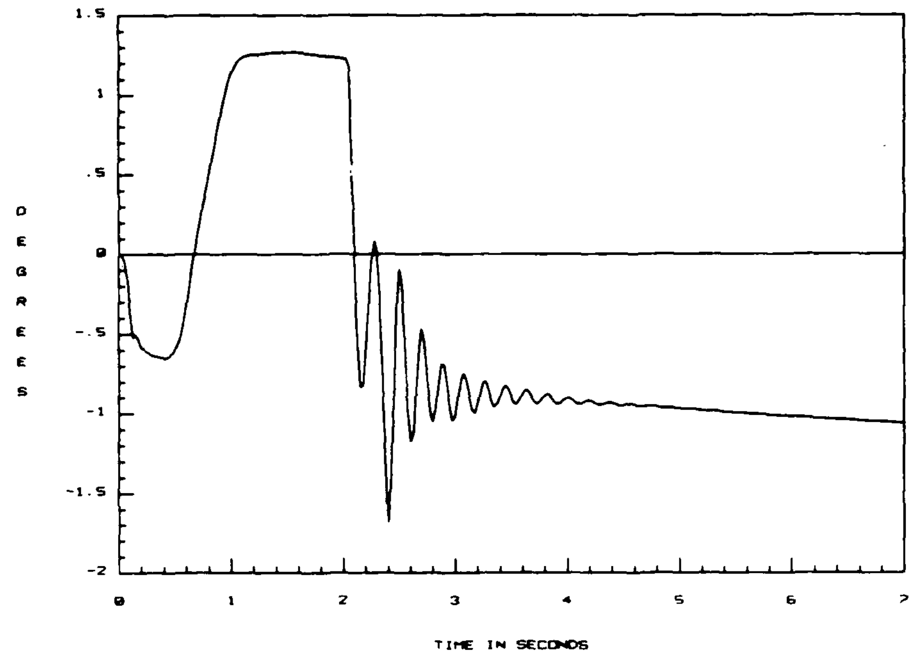


Figure 6.13. Right Canard Deflection - 45° Banked Turn

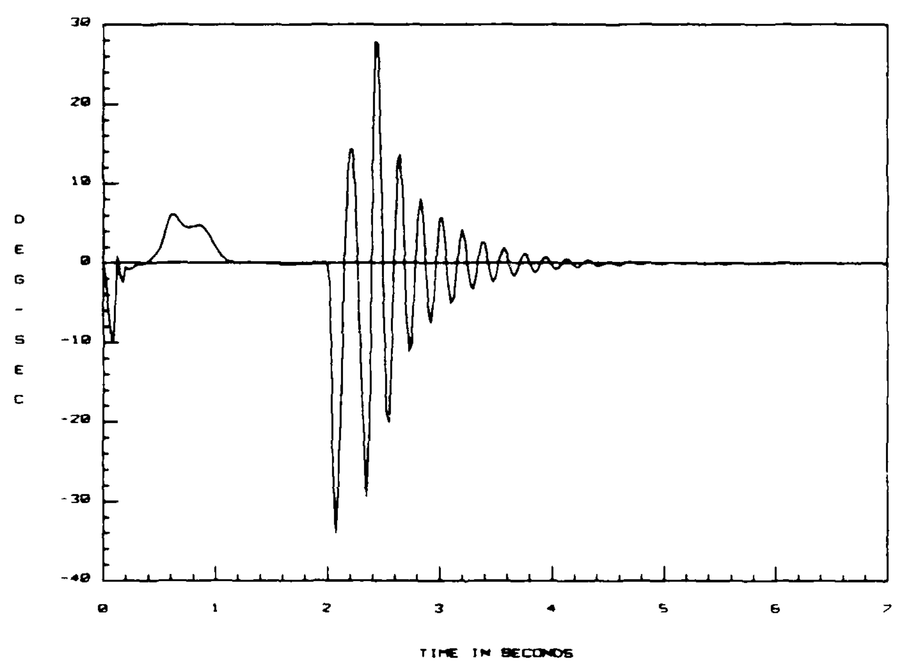


Figure 6.14. Right Canard Deflection Rate - 45° Banked Turn

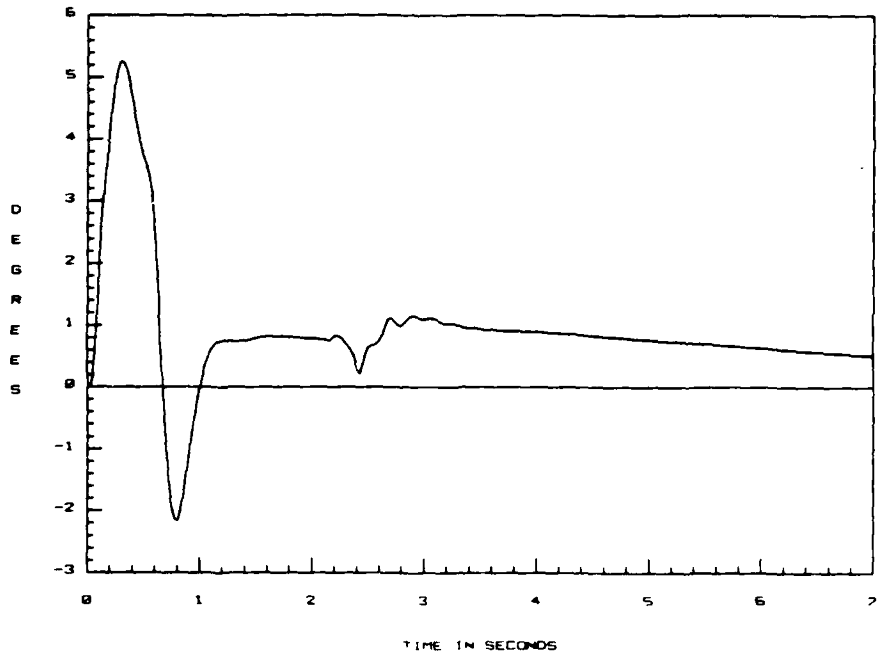


Figure 6.15. Left Trailing Edge Deflection - 45° Banked Turn

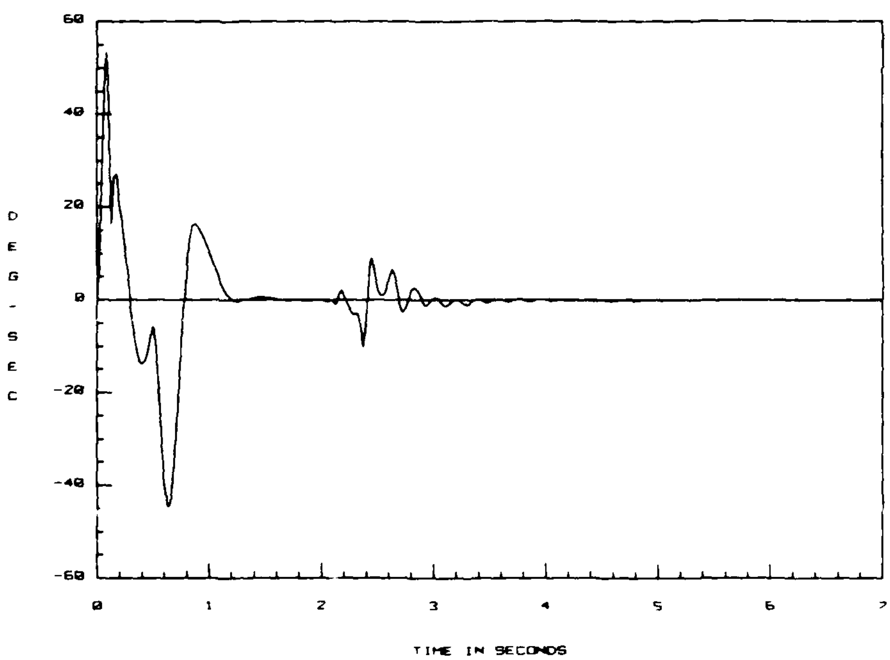


Figure 6.16. Left Trailing Edge Deflection Rate - 45° Banked Turn

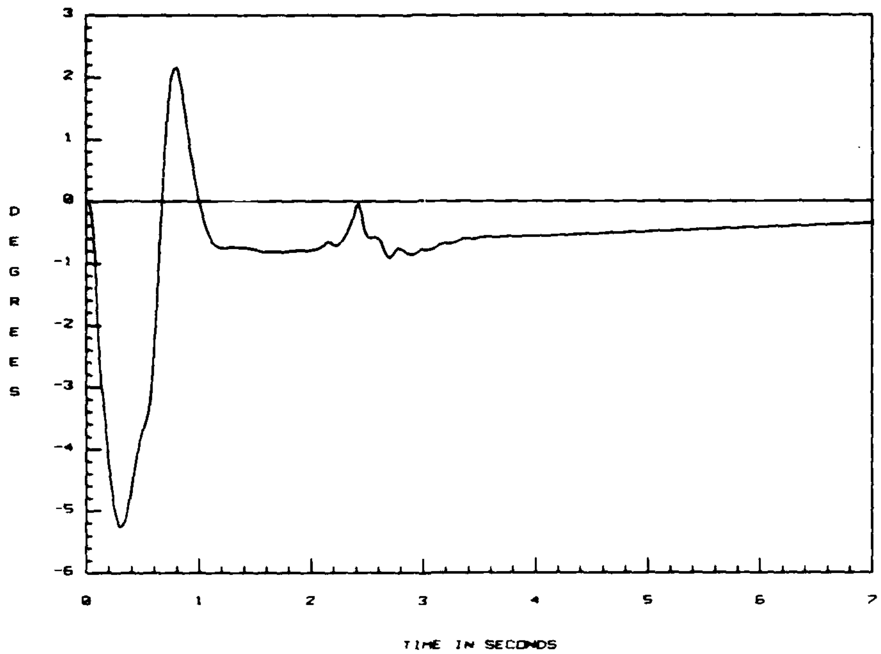


Figure 6.17. Right Trailing Edge Deflection - 45° Banked Turn

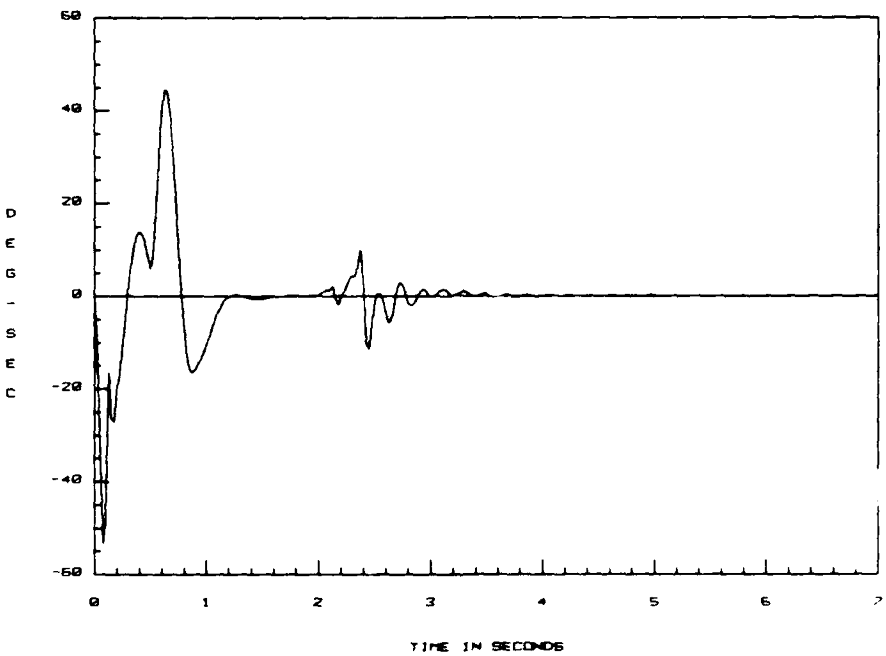


Figure 6.18. Right Trailing Edge Deflection Rate - 45° Banked Turn

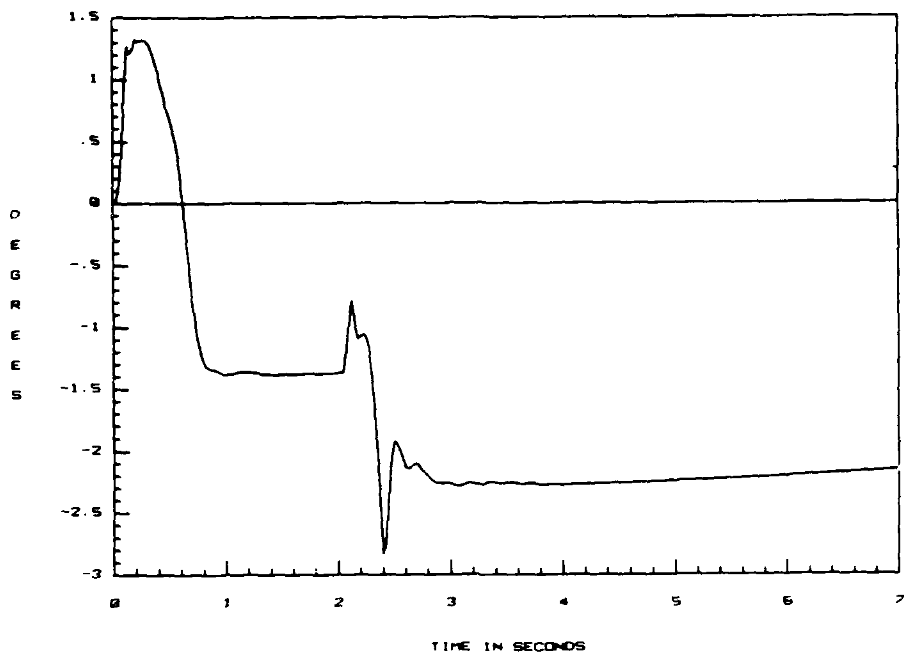


Figure 6.19. Rudder Deflection - 45° Banked Turn

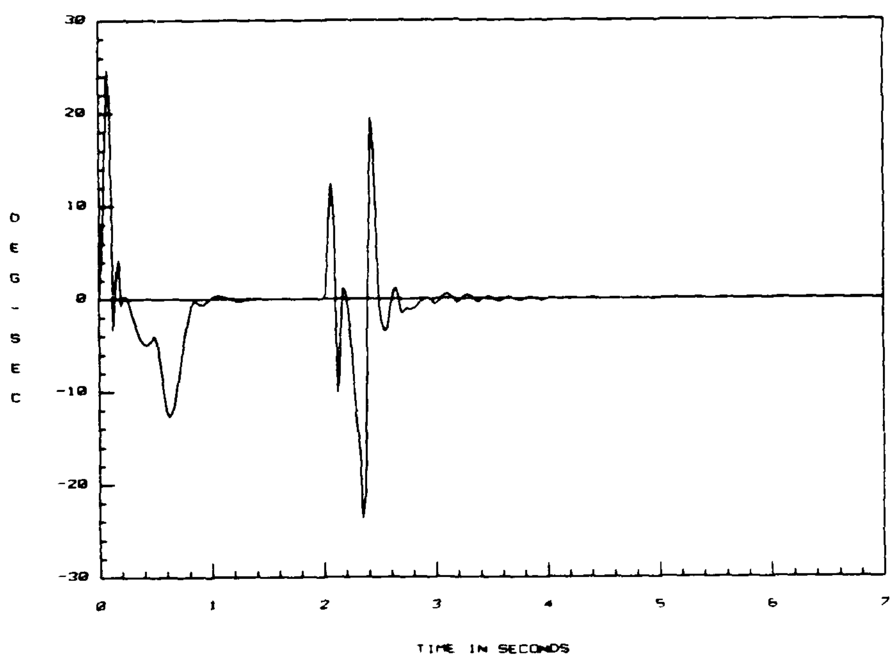


Figure 6.20. Rudder Deflection Rate - 45° Banked Turn

Time history graphs of all the plant parameter estimates are contained in Appendix C. Convergence of the parameters indicates stability, however there are fluctuations in many of the parameter values. These changes are often small in magnitude and do not significantly influence the control law gain calculations. The weighting factor α_t is adjusted to obtain the desired rate of adaption and corresponding output response.

The MATRIX$_x$ simulation accesses the RLS FORTRAN program each sample period which updates the control law before the beginning of the next sample period. The controller performs under these ideal circumstances, but an important question is how much "real" time does it take to go through the algorithm. Off-line tests of the RLS FORTRAN program indicate approximately 0.072 seconds, or about three sample periods, of computer time (VAX 8650) are needed to perform the updating of the parameter vector and control law calculations. To simulate the effects of the actual time needed to update the control law, a computational delay of three sample periods is introduced between the RLS program and the PI controller. With this delay, output responses remain stable with little or no degradation in performance. Consequently, the computational delay associated with the RLS program does not affect the performance of the controller. Because the responses are so similar, separate response plots are not included in this thesis. The delay location is shown in Figure 6.21.

6.4 Summary

When the aircraft dynamics are expressed in the form of a discrete ARMA representation and the PI control law gain calculations are based on these values, an adaptive controller implementation is feasible. The recursive least squares parameter estimation algorithm, in its simplicity of implementation, demonstrates exceptional compatability with the PI control law. A forgetting factor adjustment, filtering of input/output data, parameter estimates, and gain calculations, or spe-

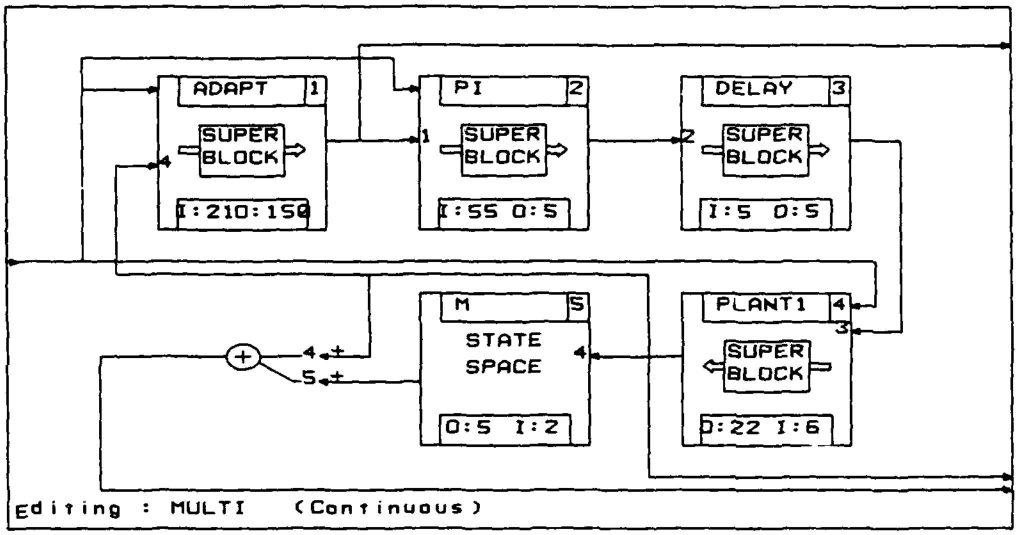


Figure 6.21. System Build Implementation of Computational Delay

cial matrix inversion FORTRAN programs have not been implemented and do not complicate the design process. Although steady state convergence of the parameter estimates are important, the individual behavior of each parameter is not an issue. First and foremost is the need to obtain the desired tracking response and aircraft stability. The next chapter presents several conclusions that have been derived from the fixed gain and adaptive controller designs of Chapters 5 and 6. Recommendations for further research are presented to examine the robustness of the multivariable design techniques.

VII. *Conclusions and Recommendations*

7.1 *Introduction*

The proportional plus integral, PI, control law developed by Professor Brian Porter of the University of Salford, England and implemented in this thesis is a result of several hundred $MATRIX_x$ simulations. Each simulation was made in hopes of adjusting design parameters for the best possible response. Fine tuning the response is inherent in any design process, but the ease of design and simulation capability of the design method is a real advantage when used with a powerful computer aided design (CAD) package like $MATRIX_x$. The tremendous power of control law design using this CAD package and the time and effort saved during the design process cannot be overemphasized. Each aspect of the design process has reinforced and contributed to the existing knowledge of PI control law design and the highlights presented in this chapter summarize the results of Chapter 5 and 6. This chapter concludes with recommendations for further research in this area of study.

7.2 *Conclusions*

For both the fixed gain and adaptive controllers, knowledge of the characteristics of the plant are needed prior to design and implementation of the PI controller. This knowledge includes controllability and observability determination, transmission zero location(s), and open-loop eigenvalues. Also a check on the control input needed to perform the desired maneuvers gives insight into the capabilities of the system. The high points of the research for the fixed gain and adaptive controller are itemized in the following sections.

7.2.1 Fixed Gain Controller 1. Criteria must be carefully chosen to evaluate the aircraft performance with the controller. This criteria provides a solid base

for controller design and should include allowable overshoot and steady state error in output responses, control surface deflection rates and limits, and determination of the steady- state control inputs needed to sustain the chosen outputs.

2. The selection of design parameters for both the continuous and discrete controllers is facilitated by use of the frequency domain analysis of the open and closed-loop responses. The discrete system design is simplified by transformation of the open and closed-loop plant and controller into the discrete W'-domain, rather than the Z-domain, where standard Bode analysis is made. In addition, the Nichols plot is an invaluable aide in pictorially illustrating gain and phase margin, especially with aircraft systems that are statically unstable and having both low and high frequency gain margins. The computer aided design package $MATRIX_x$ assists the iterative process for selection of the design variables. However, careful analysis of the phase angle data is necessary to insure that the CAD program is properly presenting the correct angle condition. $MATRIX_x$ often adds or subtracts 360 degrees from phase angle data to provide a continuous plot, which distorts the actual results and complicates controller performance evaluation.

3. The capability of the controller to maintain the desired tracking response and decoupled output in all flight conditions is exceptional! Minimizing the coupling of the output insures the fast tracking characteristics of the PI controller are optimized. Minimal cross-coupling occurs when a 50 percent loss of effectiveness of the left canard is simulated in the ACM Entry and TF/TA flight conditions. However, in all cases the steady state values are less than two degrees and 2 ft/sec for angle and velocity changes respectively. These are well within the permissible bounds.

4. The addition of the 3rd order model following input filter significantly reduces the high frequency spikes in the output and decreases the surface deflection positions and rates. Furthermore, tracking fidelity is maintained with the overall response being much smoother. Also, the closed-loop system bandwidth is reduced

from approximately 15 rad/sec to less than 10 rad/sec.

5. The discrete controller design provides output responses that are, in general, better than the continuous time system. A sampling frequency of 40 Hz used in the controller did not degrade controller performance when compared to an 80 Hz design. The similarity in developing continuous and discrete designs is one of the many advantages of the design method. In the discrete case, the controller gains K_1 and K_2 are calculated using both state- space relationships and the step-response matrix plant characteristics with appropriate weighting matrices. Although the plant is statically unstable and conventional open-loop step response matrix testing is not possible, the plant can be represented by an autoregressive difference equation with the coefficients of this equation used to determine the step response matrix. In general, the PI controller responses obtained using the gains calculated from this method were much better than either the continuous time design or the discrete controller using gains calculated from state-space relationships. Appropriate selection of the weighting matrices for the fixed gain controller using the step response matrix method simplifies the adaptive design technique by fixing the controller design parameters.

6. The ultimate goal of the PI control law design is to find a set of gains capable of satisfactory performance over the entire flight envelope. In reality this is doubtful, but a set of gains calculated at a high $\bar{q}$ condition can often provide a stable response at other flight conditions. For this research, the controller gains, K_1 and K_2 calculated at the TF/TA flight condition performed satisfactorily for all failures considered at this part of the flight envelope and provided a stable response for 66 percent of the remaining flight conditions and aircraft configurations.

7.2.2 Adaptive Controller 1. As flight conditions and aircraft configurations change, the plant properties (parameters) change to reflect the new characteristics of the plant. These changes in the plant are observed using input/output

data. Consequently, the PI control law gains can be adjusted as these changes occur as they are calculated from the current representation of the plant. The use of the autoregressive moving average (ARMA) discrete difference equation to represent the plant parameters provides a 2nd order difference equation with 100 plant parameters. Using this reduced order difference equation provide by the ARMA representation allows a recursive least squares (RLS) parameter estimation scheme that estimates all plant parameters.

2. The estimating scheme is standard RLS without consideration of a forgetting factor. The rate of adaption is controlled via the noise variance term, α_t, and immensely simplifies the design process. Since the controller weighting matrices are established from the fixed gain design, only one design parameter is adjusted. Selecting the proper value of α_t is an iterative process, but it is simplified by using MATRIX_x . Double precision matrix inversion FORTRAN subroutines performed quite well and eliminated the need for U-D factorization required when ill-conditioned matrices exist.

3. The selected failure to test the RLS adaption is a 30 percent loss of effectiveness of the left trailing edge which fell into the approximate 30 percent of the flight conditions that could *not* be used with a universal gain controller. The failure was introduced two seconds into the simulation and the rate of adaption and reestablishment of control was extraordinary. The response settles back to steady state within three seconds after the failure is introduced with control surface deflections and rates well within normal. Readjustment of the controller gains resulted in good performance in all cases.

4. In most cases, the tracking of plant parameters is not a concern. Time history graphs of plant parameter values indicate no divergence in parameter estimates and steady-state values reached. However, during the adaption process fluctuations in individual estimates occur. Since the gain calculations are based upon different linear combinations of all the parameters, erratic behavior of the

parameters does not have a significant affect on the gain calculations. Typically, the fluctuations are of small magnitude. Of prime importance is the stability and response of the output, not how each parameter is changing!

5. The actual time to perform the recursive least squares estimation in the FORTRAN program is about 0.072 seconds (VAX 8650), or three sample periods. Since the simulation updates the control law each sample period, a means to evaluate the controller for the actual time is necessary. $MATRIX_x$ has the capability to simulate the effects of this computational delay. Simulation results with a three sample period delay introduced in the computational loop show little degradation in output response.

7.3 Recommendations for Further Study

The CRCA provides a unique test bed for the development of a suitable control law because of the complexity of aerodynamic cross-coupling. This thesis presents an initial attempt to demonstrate the effectiveness of the fixed gain and adaptive controller at several points in the flight envelope. Simplicity in the implementation of the recursive least squares estimation of plant parameters demonstrates an ability to accomplish on-line estimation and control law updating. However, certain issues and concerns need to be investigated and evaluated to further reinforce the robustness of the design technique. The following list offers areas of interest.

1. The PID control law can eliminate the need for the measurement matrix, M, when an *irregular* plant configuration exists. Thus, the slow modes of the system introduced by the measurement matrix are eliminated. A PID adaptive controller implementation is recommended after first stabilizing the plant. Demonstration of an acceptable PID controller design and ease of design for both fixed gain and adaptive control needs to be accomplished.

2. The simulation for the adaptive controller should include the effects of

noise and other disturbances. Persistent excitation and a sufficiently rich input aides the adaption process and the intentional introduction of some sensor noise may improve the response, but a simulation under varied conditions should be checked. In the event noise degrades the performance of the controller, application of Kalman filtering techniques may provide a strong solution.

3. The rate of adaption and stable output response is adjusted using the noise variance term, α_t, in the RLS algorithm. Although, this simplicity of design performed satisfactorily, investigation of alternative ways to tailor the output response is necessary. Filtering of input/output data entering the adaptive algorithm or filtering the gain estimates generated by the adaptive algorithm are other ways to adjust the response and may provide a viable option.

4. The controller and associated aircraft dynamics should be simulated on an analog computer, such as the SIMSTAR, in real time. In this case, the Neal-Smith Criterion or the Cooper-Harper flying quality ratings could be applied to the controller design with pilot in the loop simulations.

Appendix A. *Aircraft Data*

Introduction

This appendix contains the aircraft data used at the flight conditions of ACM Entry, ACM Exit, and TF/TA. Included are the stability coefficients used in the Flight Dynamics Lab computer program LEMCO, Linearized Equations of Motion, and the corresponding output of the A, B and C state space matrices used in the design. The appendix concludes with a listing of the equivalent ARMA difference equation coefficients at each linearized flight condition. The information contained in Table A.1 through Table A.7 was obtained from the Flight Dynamics Lab CRCA support personnel.

Stability Coefficients and State Space Models

The Flight Dynamics Lab at Wright-Patterson AFB, in conjunction with Grumman and Lear Corporation, has provided a comprehensive listing of the aircraft stability coefficients at each of the flight conditions desired. Table A.1 through Table A.7 provide reproduction of this data for the values representing the aircraft at the analyzed flight condition. The program LEMCO performed the linearization of the equations of motion at the equilibrium condition using these coefficient values. The program output provides complete state space A and B matrices. Table A.8 through Table A.14 reflect the output generated by LEMCO, but rearranged in the form necessary for the Porter control law analysis. The state variables are those values listed in Chapter 2.

Table A.1. ACM Entry Aero Data - No Failures

Flight Condition

M	=	.9	$\bar{q}$	=	356.3 lb/ft^2
Alt	=	30,000 ft	Thrust	=	6134 lb
V	=	895 ft/sec	DLFLP	=	0.0 deg

Aircraft Attitude

α	=	2.02 deg	θ	=	2.02 deg
β	=	0.00 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-5.09	TE1L	=	2.586
CR	=	-5.09	TE1R	=	2.586
DTE1SL	=	2.586	TE2L	=	2.586
DTE1SR	=	2.586	TE2R	=	2.586
RUDR	=	0.0			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0063 (deg^{-1})	C_{mq}	=	-3.4000 (rad^{-1})
$C_{n\alpha}$	=	.0761 (deg^{-1})	C_{nu}	=	-.0110 $(mach^{-1})$
$C_{a\alpha}$	=	-.0039 (deg^{-1})			

Lateral/Directional

$C_{y\beta}$	=	-.0067 (deg^{-1})	C_{np}	=	-.0290 (rad^{-1})
C_{lr}	=	.0610 (rad^{-1})	$C_{l\beta}$	=	.0014 (deg^{-1})
C_{lp}	=	-.2530 (rad^{-1})	C_{nr}	=	-.5270 (rad^{-1})
$C_{n\beta}$	=	.0012 (deg^{-1})			

Table A.1 ACM Entry Aero Data - No Failures (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00588	.00146	.00019	.00132	.00022	.00136
Right Canard	.00588	.00146	.00019	-.00132	-.00022	-.00136
L Obd Flaperon	-.00125	.00286	.00040	-.00043	.00053	.00006
L Inbd Flaperon	-.00165	.00443	.00061	-.00067	.00064	.00008
Left Elevator	-.00116	.00338	.00047	-.00051	.00032	.00006
Right Elevator	-.00116	.00338	.00047	.00051	-.00032	-.00006
R Inbd Flaperon	-.00165	.00443	.00061	.00067	-.00064	-.00008
R Obd Flaperon	-.00125	.00286	.00040	.00043	-.00058	-.00006
Rudder	0	0	0	.00255	.00033	-.00152

Table A.2. ACM Entry Aero Data - 30 Percent Loss of Effectiveness TEL

Flight Condition

M	=	.9	$\bar{q}$	=	356.3 lb/ft^2
Alt	=	30,000 ft	Thrust	=	5431 lb
V	=	895 ft/sec	DLFLP	=	-2.5 deg

Aircraft Attitude

α	=	2.56 deg	θ	=	2.56 deg
β	=	0.00 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-7.025	TE1L	=	1.699
CR	=	-7.024	TE1R	=	.990
DTE1SL	=	1.699	TE2L	=	1.699
DTE1SR	=	.990	TE2R	=	.990
RUDR	=	.001			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0063 (deg^{-1})	C_{mq}	=	-3.5500 (rad^{-1})
$C_{n\alpha}$	=	.0739 (deg^{-1})	C_{nu}	=	-.0110 $(mach^{-1})$
$C_{a\alpha}$	=	-.0040 (deg^{-1})			

Lateral/Directional

$C_{y\beta}$	=	-.0071 (deg^{-1})	C_{np}	=	-.0370 (rad^{-1})
C_{lr}	=	.0640 (rad^{-1})	$C_{l\beta}$	=	-.0016 (deg^{-1})
C_{lp}	=	-.2650 (rad^{-1})	C_{nr}	=	-.5310 (rad^{-1})
$C_{n\beta}$	=	.0013 (deg^{-1})			

Table A.2 ACM Entry Aero Data - 30 Percent Loss of Effectiveness TEL (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00589	.00152	.00024	.00134	.00023	.00137
Right Canard	.00589	.00152	.00024	-.00137	-.00022	-.00135
L Obd Flaperon	-.00124	.00287	.00036	-.00055	.00058	.00008
L Inbd Flaperon	.00000	.00000	.00000	.00000	.00000	.00000
Left Elevator	-.00115	.00339	.00043	-.00065	.00032	.00007
Right Elevator	-.00115	.00339	.00037	.00065	-.00032	-.00007
R Inbd Flaperon	-.00164	.00445	.00048	.00085	-.00064	-.00010
R Obd Flaperon	-.00124	.00287	.00031	.00055	-.00058	-.00008
Rudder	0	0	0	.00257	.00033	-.00154

Table A.3. ACM Entry Aero Data - 50 Percent Loss of Effectiveness CL

Flight Condition

M	=	.9	$\bar{q}$	=	356.3 lb/ft^2
Alt	=	30,000 ft	Thrust	=	5909 lb
V	=	895 ft/sec	DLFLP	=	-2.5 deg

Aircraft Attitude

α	=	2.30 deg	θ	=	2.30 deg
β	=	-1.44 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-6.452	TE1L	=	1.686
CR	=	-7.655	TE1R	=	1.728
DTE1SL	=	1.686	TE2L	=	1.686
DTE1SR	=	.990	TE2R	=	1.728
RUDR	=	-1.204			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0038 (deg^{-1})	C_{mq}	=	-3.5000 (rad^{-1})
$C_{n\alpha}$	=	.0753 (deg^{-1})	C_{nu}	=	-.0110 $(mach^{-1})$
$C_{a\alpha}$	=	-.0041 (deg^{-1})			

Lateral/Directional

$C_{y\beta}$	=	-.0075 (deg^{-1})	C_{np}	=	.0330 (rad^{-1})
C_{lr}	=	.0630 (rad^{-1})	$C_{l\beta}$	=	-.0015 (deg^{-1})
C_{lp}	=	-.2590 (rad^{-1})	C_{nr}	=	-.5250 (rad^{-1})
$C_{n\beta}$	=	.0015 (deg^{-1})			

Table A.3 ACM Entry Aero Data - 50 Percent Loss of Effectiveness CL (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00294	.00075	.00022	.00070	.00014	.00068
Right Canard	.00588	.00150	.00022	-.00061	-.00009	-.00068
L Obd Flaperon	-.00125	.00287	.00037	-.00049	.00058	.00007
L Inbd Flaperon	-.00165	.00444	.00057	-.00076	.00064	.00009
Left Elevator	-.00115	.00338	.00043	-.00058	.00032	.00006
Right Elevator	-.00115	.00338	.00043	.00058	-.00032	-.00006
R Inbd Flaperon	-.00165	.00444	.00057	.00076	-.00064	-.00009
R Obd Flaperon	-.00125	.00287	.00037	.00049	-.00058	-.00007
Rudder	0	0	0	.00256	.00033	-.00153

Table A.4. ACM Exit Aero Data - No Failures

Flight Condition

M	= .275	$\bar{q}$	= 77.0 lb/ft^2
Alt	= 10,000 ft	Thrust	= 47092 lb
V	= 296 ft/sec	DLFLP	= 30.0 deg

Aircraft Attitude

α	= 2.56 deg	θ	= 5.74 deg
β	= 3.59 deg	ϕ	= -70.53 deg

Surface Positions (deg)

CL	= -19.17	TE1L	= 19.74
CR	= -19.90	TE1R	= 11.40
DTE1SL	= 19.70	TE2L	= 19.74
DTE1SR	= 11.40	TE2R	= 11.40
RUDR	= -.727		

Stability Derivatives

Longitudinal

$C_{m\alpha}$	= .0054 (deg^{-1})	C_{mq}	= -5.6300 (rad^{-1})
$C_{n\alpha}$	= .0587 (deg^{-1})	C_{nu}	= -.0110 $(mach^{-1})$
$C_{a\alpha}$	= .0006 (deg^{-1})		

Lateral/Directional

$C_{y\beta}$	= -.0027 (deg^{-1})	C_{np}	= -.1000 (rad^{-1})
C_{lr}	= .1840 (rad^{-1})	$C_{l\beta}$	= -.0018 (deg^{-1})
C_{lp}	= -.2760 (rad^{-1})	C_{nr}	= .0840 (rad^{-1})
$C_{n\beta}$	= .0002 (deg^{-1})		

Table A.4 ACM Exit Aero Data - No Failures (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00431	.00323	-.00054	.00467	.00015	.00172
Right Canard	.00418	.00314	-.00060	-.00443	-.00045	-.00156
L Obd Flaperon	-.00048	.00181	.00028	-.00042	.00032	-.00005
L Inbd Flaperon	-.00064	.00278	.00044	-.00065	.00036	-.00007
Left Elevator	-.00044	.00213	.00033	-.00050	.00018	-.00005
Right Elevator	-.00044	.00213	.00033	.00050	-.00018	.00005
R Inbd Flaperon	-.00064	.00278	.00044	.00066	-.00036	.00007
R Obd Flaperon	-.00048	.00181	.00028	.00043	-.00032	.00005
Rudder	0	0	0	.00282	.00037	-.00168

Table A.5. TF/TA Aero Data - No Failures

Flight Condition

M	=	.9	$\bar{q}$	=	1200.0 lb/ft^2
Alt	=	Sea Level	Thrust	=	19180 lb
V	=	1005 ft/sec	DLFLP	=	0.0 deg

Aircraft Attitude

α	=	0.79 deg	θ	=	0.79 deg
β	=	0.00 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-4.695	TE1L	=	3.718
CR	=	-4.695	TE1R	=	3.718
DTE1SL	=	3.718	TE2L	=	3.718
DTE1SR	=	3.718	TE2R	=	3.718
RUDR	=	.000			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0061 (deg^{-1})	C_{mq}	=	-3.1600 (rad^{-1})
$C_{n\alpha}$	=	.0766 (deg^{-1})	C_{nu}	=	-.0110 $(mach^{-1})$
$C_{a\alpha}$	=	.0001 (deg^{-1})			

Lateral/Directional

$C_{y\beta}$	=	-.0061 (deg^{-1})	C_{np}	=	-.0120 (rad^{-1})
C_{lr}	=	.0540 (rad^{-1})	$C_{l\beta}$	=	-.0010 (deg^{-1})
C_{lp}	=	-.2260 (rad^{-1})	C_{nr}	=	-.5100 (rad^{-1})
$C_{n\beta}$	=	.0012 (deg^{-1})			

Table A.5 TF/TA Aero Data - No Failures (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00411	.00093	-.00045	.00170	.00020	.00132
Right Canard	.00411	.00093	-.00045	-.00170	-.00020	-.00132
L Obd Flaperon	-.00070	.00155	.00023	-.00009	.00032	.00001
L Inbd Flaperon	-.00092	.00240	.00035	-.00014	.00036	.00002
Left Elevator	-.00064	.00183	.00027	-.00011	.00018	.00001
Right Elevator	-.00064	.00183	.00027	.00011	-.00018	-.00001
R Inbd Flaperon	-.00092	.00240	.00035	.00014	-.00036	-.00002
R Obd Flaperon	-.00070	.00155	.00023	.00009	-.00032	-.00001
Rudder	0	0	0	.00250	.00032	-.00149

Table A.6. TF/TA Aero Data - 30 Percent Loss of Effectiveness TEL

Flight Condition

M	=	.9	$\bar{q}$	=	1200.0 lb/ft^2
Alt	=	Sea Level	Thrust	=	18839 lb
V	=	1005 ft/sec	DLFLP	=	0.0 deg

Aircraft Attitude

α	=	0.91 deg	θ	=	0.91 deg
β	=	0.00 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-5.100	TE1L	=	5.022
CR	=	-5.099	TE1R	=	2.926
DTE1SL	=	5.022	TE2L	=	5.022
DTE1SR	=	2.926	TE2R	=	2.926
RUDR	=	.000			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0062 (deg^{-1})	C_{mq}	=	-3.1700 (rad^{-1})
$C_{n\alpha}$	=	.0747 (deg^{-1})	C_{nu}	=	-.0110 ($mach^{-1}$)
$C_{a\alpha}$	=	-.0005 (deg^{-1})			

Lateral/Directional

$C_{y\beta}$	=	-.0060 (deg^{-1})	C_{np}	=	-.0140 (rad^{-1})
C_{lr}	=	.0540 (rad^{-1})	$C_{l\beta}$	=	-.0010 (deg^{-1})
C_{lp}	=	-.2280 (rad^{-1})	C_{nr}	=	-.5140 (rad^{-1})
$C_{n\beta}$	=	.0012 (deg^{-1})			

Table A.6 TF/TA Aero Data - 30 Percent Loss of Effectiveness TEL (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00412	.00094	-.00048	.00174	.00020	.00132
Right Canard	.00412	.00094	-.00048	-.00174	-.00020	-.00132
L Obd Flaperon	-.00070	.00155	.00022	-.00011	.00032	.00001
L Inbd Flaperon	.00000	.00000	.00013	.00000	.00000	.00000
Left Elevator	-.00064	.00183	.00027	-.00013	.00018	.00001
Right Elevator	-.00064	.00183	.00027	.00013	-.00018	-.00001
R Inbd Flaperon	-.00092	.00240	.00035	.00016	-.00036	-.00002
R Obd Flaperon	-.00070	.00155	.00022	.00011	-.00032	-.00001
Rudder	0	0	0	.00250	.00033	-.00149

Table A.7. TF/TA Aero Data - 50 Percent Loss of Effectiveness CL

Flight Condition

M	=	.9	$\bar{q}$	=	1200.0 lb/ft^2
Alt	=	Sea Level	Thrust	=	20021 lb
V	=	1005 ft/sec	DLFLP	=	0.0 deg

Aircraft Attitude

α	=	0.67 deg	θ	=	0.67 deg
β	=	-0.01 deg	ϕ	=	0.00 deg

Surface Positions (deg)

CL	=	-4.964	TE1L	=	4.639
CR	=	-5.515	TE1R	=	4.177
DTE1SL	=	4.639	TE2L	=	4.639
DTE1SR	=	4.177	TE2R	=	4.177
RUDR	=	-.551			

Stability Derivatives

Longitudinal

$C_{m\alpha}$	=	.0036 (deg^{-1})	C_{mq}	=	-3.1300 (rad^{-1})
$C_{n\alpha}$	=	.0760 (deg^{-1})	C_{nu}	=	-.0110 $(mach^{-1})$
$C_{a\alpha}$	=	.0005 (deg^{-1})			

Lateral Directional

$C_{y\beta}$	=	-.0061 (deg^{-1})	C_{np}	=	-.0110 (rad^{-1})
C_{lr}	=	.0530 (rad^{-1})	$C_{l\beta}$	=	-.0010 (deg^{-1})
C_{lp}	=	-.2230 (rad^{-1})	C_{nr}	=	-.5100 (rad^{-1})
$C_{n\beta}$	=	.0012 (deg^{-1})			

Table A.7 TF/TA Aero Data- 50 Percent Loss of Effectiveness CL (cont)

Control Derivatives (deg^{-1})

Surface	$C_{m\delta}$	$C_{n\delta}$	$C_{a\delta}$	$C_{y\delta}$	$C_{l\delta}$	$C_{n\delta}$
Left Canard	.00206	.00046	-.00048	.00094	.00011	.00065
Right Canard	.00414	.00093	-.00054	-.00000	-.00009	-.00066
L Obd Flaperon	-.00070	.00155	.00023	-.00008	.00032	.00001
L Inbd Flaperon	-.00092	.00240	.00035	-.00012	.00036	.00001
Left Elevator	-.00065	.00183	.00027	-.00009	.00018	.00001
Right Elevator	-.00065	.00183	.00027	.00009	-.00018	-.00001
R Inbd Flaperon	-.00092	.00240	.00035	.00012	-.00036	-.00001
R Obd Flaperon	-.00070	.00155	.00023	.00008	-.00032	-.00001
Rudder	0	0	0	.00249	.00032	-.00149

Table A.8. ACM Entry Matrices - No Failures

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0350 \\ -32.1804 & .0000 & -.0119 & -.0186 & -31.2350 & .0000 & .0000 & .0000 \\ -1.0634 & .0000 & -.0324 & -1.0634 & 894.4548 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0069 & -.6015 & .0000 & .0000 & .0000 \\ .0000 & .0360 & .0000 & .0000 & .0000 & -.0929 & .0349 & -.9994 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -27.8066 & -2.0376 & .4913 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 2.4582 & -.0241 & -.4377 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0010 \\ .0411 & .0411 & .1322 & .0866 & .1322 & .0866 & .1018 & .1018 & .0000 \\ -.3163 & -.3163 & -.9597 & -.6194 & -.9597 & -.6194 & -1.0183 & -1.0183 & .0000 \\ .1014 & .1014 & -.0284 & -.0215 & -.0284 & -.0215 & -.0200 & -.0200 & .0000 \\ .0003 & -.0003 & -.0002 & -.0001 & .0002 & .0001 & -.0001 & .0001 & .0006 \\ .0762 & -.0762 & .2219 & .2011 & -.2219 & -.2011 & .1109 & -.1109 & .1144 \\ .0486 & -.0486 & .0029 & .0021 & -.0029 & -.0021 & .0021 & -.0021 & -.0544 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .0349 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

B Matrix for Five Control Surfaces

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0411 & .0411 & .3206 & .3206 & .0000 \\ -.3163 & -.3163 & -2.5974 & -2.5974 & .0000 \\ .1014 & .1014 & -.0699 & -.0699 & .0000 \\ .0003 & -.0003 & -.0004 & .0004 & .0006 \\ .0762 & -.0762 & .5339 & -.5339 & .1144 \\ .0486 & -.0486 & .0071 & -.0071 & -.0544 \end{bmatrix}$$

$$x = \left[\theta \quad \phi \quad u \quad w \quad q \quad \beta \quad p \quad r \right]^T \tag{A.1}$$

$$u = \left[\delta_{cl} \quad \delta_{cr} \quad \delta_{ter} \quad \delta_{tel} \quad \delta_{rud} \right]^T \tag{A.2}$$

Table A.9. ACM Entry Matrices - 30 Percent Loss Of Effectiveness TEL

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0450 \\ -32.1420 & .0000 & -.0050 & .0550 & -39.9760 & .0000 & .0000 & .0000 \\ -1.4370 & .0000 & -.0240 & -1.0280 & 894.1070 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0070 & -.6920 & .0000 & .0000 & .0000 \\ .0000 & .0360 & .0000 & .0000 & .0000 & -.0990 & .0450 & -.9990 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -31.8340 & -2.1380 & .5160 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 2.6670 & -.0310 & -.4420 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0520 & .0520 & .0000 & .0780 & .1040 & .0670 & .0930 & .0800 & .0000 \\ -.3300 & -.3300 & .0000 & -.6220 & -.9650 & -.6220 & -.7350 & -.7350 & .0000 \\ .1020 & .1020 & .0000 & -.0210 & -.0280 & -.0210 & -.0200 & -.0200 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0010 \\ .0800 & -.0760 & .0000 & .2010 & -.2220 & -.2010 & .1110 & -.1110 & .1150 \\ .0490 & -.0480 & .0000 & .0030 & -.0040 & -.0030 & .0030 & -.0030 & .0550 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .0410 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & 0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

<u>B Matrix for Five Control Surfaces</u>

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0520 & .0520 & .1710 & .2510 & .0000 \\ -.3303 & -.3300 & -1.3570 & -2.3220 & .0000 \\ .1020 & .1020 & -.0410 & -.0690 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0010 \\ .0800 & -.0760 & .3120 & -.5340 & .1150 \\ .0490 & -.0480 & .0060 & -.0100 & .0550 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \qquad (A.3)$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \qquad (A.4)$$

Table A.10. ACM Entry Matrices - 50 Percent Loss Of Effectiveness CL

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0390 \\ -32.1500 & .0000 & -.0080 & .0580 & -36.6690 & .0000 & .0000 & .0000 \\ -1.2460 & .0000 & -.0290 & -1.0450 & 894.3280 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0040 & -.6460 & .0000 & .0000 & .0000 \\ .0000 & .0360 & .0000 & .0000 & .0000 & -.0820 & .0390 & -.9990 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -23.8380 & -2.0700 & .4990 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 2.4580 & -.0270 & -.4520 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ -.1560 & -.1150 & .1320 & .0840 & .1320 & .0840 & .1000 & .1000 & .0000 \\ -.1150 & -.2250 & -.9610 & -.6210 & -.9610 & -.6210 & -.7320 & -.7320 & .0000 \\ .0360 & .0720 & -.0280 & -.0220 & -.0280 & -.0220 & -.0200 & -.0200 & .0000 \\ .0000 & -.0010 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0010 \\ .0350 & -.0690 & .2220 & .2010 & -.2220 & -.2010 & .1110 & -.1110 & .1140 \\ .0240 & -.0490 & .0030 & .0030 & -.0030 & -.0030 & .0020 & -.0020 & -.0550 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .0349 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

B Matrix for Five Control Surfaces

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ -.1560 & -.1150 & .3160 & .3160 & .0000 \\ -.1150 & -.2250 & -2.3140 & -2.3140 & .0000 \\ .0360 & .0720 & -.0700 & -.0700 & .0000 \\ .0000 & -.0010 & .0000 & .0000 & .0010 \\ .0350 & -.0690 & .5340 & -.5340 & .1140 \\ .0240 & -.0490 & .0080 & -.0080 & -.0550 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \quad (A.5)$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \quad (A.6)$$

Table A.11. ACM Exit Matrices - No Failures

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .5184 \\ -28.5877 & .0000 & -.2762 & -.0283 & -136.2190 & .0000 & .0000 & .0000 \\ -4.9465 & 26.9479 & .1330 & -.6753 & 262.7934 & .0000 & .0000 & .0000 \\ .0000 & .0000 & -.0018 & .0035 & -.6511 & .0000 & .0000 & .0000 \\ .0472 & .0322 & .0000 & .0000 & .0000 & -.0245 & .4602 & -.8878 \\ .0000 & .0000 & 0000 & .0000 & .0000 & -7.7280 & -1.4530 & .9687 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -.0889 & -.0543 & .0456 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ -.0253 & -.0281 & .0206 & .0131 & .0206 & .0131 & .0154 & .0154 & .0000 \\ -.1512 & -.1471 & -.1302 & -.0848 & -.1302 & -.0848 & -.0997 & -.0997 & .0000 \\ .0161 & -.0156 & -.0024 & -.0018 & -.0024 & -.0018 & -.0016 & -.0016 & .0000 \\ .0007 & -.0007 & -.0001 & -.0001 & .0001 & .0001 & -.0001 & .0001 & .0004 \\ .0113 & -.0337 & .0270 & .0240 & -.0270 & -.0240 & .0135 & -.0135 & .0277 \\ .0133 & -.0121 & -.0005 & -.0004 & .0005 & .0004 & -.0004 & .0004 & -.0130 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .4600 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

B Matrix for Five Control Surfaces

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ -.0253 & -.0281 & .0491 & .0491 & .0000 \\ -.1512 & -.1471 & -.3147 & -.3147 & .0000 \\ .0161 & .0156 & -.0058 & -.0058 & .0000 \\ .0007 & -.0007 & -.0003 & .0003 & .0004 \\ .0113 & -.0337 & .0645 & -.0645 & .0277 \\ .0133 & -.0121 & -.0013 & .0013 & -.0130 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \tag{A.7}$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \tag{A.8}$$

Table A.12. TF/TA Matrices - No Failures

$$
A = \begin{bmatrix}
.0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\
.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0153 \\
-32.1961 & .0000 & -.0355 & .0357 & -15.6105 & .0000 & .0000 & .0000 \\
-.5002 & .0000 & -.0071 & -3.2056 & 1004.8788 & .0000 & .0000 & .0000 \\
.0000 & .0000 & -.0003 & .0202 & -1.6773 & .0000 & .0000 & .0000 \\
.0000 & .0320 & .0000 & .0000 & .0000 & -.2538 & .0155 & -.9999 \\
.0000 & .0000 & .0000 & .0000 & .0000 & -66.9300 & -5.4612 & 1.0349 \\
.0000 & .0000 & .0000 & .0000 & .0000 & 8.2821 & -.0299 & -1.2709
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\
.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\
-.3284 & -.3284 & .2548 & .1679 & .2548 & .1679 & .1975 & .1975 & .0000 \\
-.6788 & -.6788 & -1.7517 & -1.1313 & -1.7517 & -1.1313 & -1.3351 & -1.3351 & .0000 \\
.2387 & .2387 & -.0534 & -.0406 & -.0534 & -.0406 & -.0372 & -.0372 & .0000 \\
.0012 & -.0012 & -.0001 & -.0001 & .0001 & .0001 & -.0001 & .0003 & .0018 \\
.2336 & -.2336 & .4200 & .3737 & -.4200 & -.3737 & .2100 & -.2100 & .3737 \\
.1590 & -.1590 & .0024 & .0012 & -.0024 & -.0012 & -.0012 & -.0012 & -.1795
\end{bmatrix}
$$

$$
C = \begin{bmatrix}
.0000 & .0000 & 1.0000 & .0138 & .0000 & .0000 & .0000 & .0000 \\
.0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\
1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\
.0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\
.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000
\end{bmatrix}
$$

<u>B Matrix for Five Control Surfaces</u>

$$
B = \begin{bmatrix}
.0000 & .0000 & .0000 & .0000 & .0000 \\
.0000 & .0000 & .0000 & .0000 & .0000 \\
-.3284 & -.3284 & .6202 & .6202 & .0000 \\
-.6788 & -.6788 & -4.2181 & -4.2181 & .0000 \\
.2387 & .2387 & -.1312 & -.1312 & .0000 \\
.0012 & -.0012 & -.0003 & .0005 & .0018 \\
.2336 & -.2336 & 1.0037 & -1.0037 & .3737 \\
.1590 & -.1590 & .0048 & -.0048 & -.1795
\end{bmatrix}
$$

$$
x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \tag{A.9}
$$

$$
u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \tag{A.10}
$$

Table A.13. TF/TA Matrices - 30 Percent Loss of Effectiveness TEL

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0160 \\ -32.1704 & .0000 & -.0330 & -.0200 & -15.2961 & .0000 & .0000 & .0000 \\ -.5110 & .0000 & -.0080 & -3.0620 & 1004.8730 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0210 & -1.8240 & .0000 & .0000 & .0000 \\ .0000 & .0320 & .0000 & .0000 & .0000 & -.2460 & .0160 & -1.0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -65.9740 & -5.4310 & 1.2860 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 8.1650 & -.0340 & -1.2630 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ -.3450 & -.3450 & .0930 & .1580 & .2520 & .1580 & .1940 & .1940 & .0000 \\ -.6760 & -.6760 & .0000 & -1.1140 & -1.7250 & -1.1140 & -1.3160 & -1.3160 & .0000 \\ .2360 & .2360 & .0000 & -.0400 & -.0520 & -.0400 & -.0360 & -.0360 & .0000 \\ .0010 & -.0010 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0020 \\ .2300 & -.2300 & .0000 & .3680 & -.4150 & -.3680 & .2070 & -.2070 & .3800 \\ .1570 & -.1570 & .0000 & .0010 & -.0020 & -.0010 & .0010 & -.0010 & -.1770 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .0160 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

B Matrix for Five Control Surfaces

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ -.3450 & -.3450 & .4450 & .6042 & .0000 \\ -.6760 & -.6760 & -2.4300 & -4.1550 & .0000 \\ .2360 & .2360 & -.0760 & -.1280 & .0000 \\ .0010 & -.0010 & .0000 & .0000 & .0020 \\ .2300 & -.2300 & .5750 & .9900 & .3800 \\ .1570 & -.1570 & .0020 & -.0040 & -.1770 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \tag{A.11}$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \tag{A.12}$$

Table A.14. TF/TA Matrices - 50 Percent Loss of Effectiveness CL

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0110 \\ -32.1720 & .0000 & -.0380 & -.0290 & -11.0560 & .0000 & .0000 & .0000 \\ -.3540 & .0000 & -.0160 & -3.6430 & 1005.4390 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0140 & -2.1070 & .0000 & .0000 & .0000 \\ .0000 & .0320 & .0000 & .0000 & .0000 & -.2780 & .0110 & -1.0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -77.1970 & -6.1840 & 1.4760 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 8.7570 & -.0290 & -1.4650 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ -.5220 & -.3790 & .2940 & .1930 & .2940 & .1930 & .2270 & .2270 & .0000 \\ -.3950 & -.7740 & -2.0190 & -1.3040 & -2.0190 & -1.3040 & -1.5390 & -1.5390 & .0000 \\ .1390 & .2750 & -.0610 & -.0470 & -.0610 & -.0470 & -.0430 & -.0430 & .0000 \\ .0010 & -.0020 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0018 \\ .1210 & -.2830 & .4850 & .4310 & -.4850 & -.4310 & .2430 & -.2430 & .4310 \\ .0920 & -.1820 & .0010 & .0010 & -.0010 & -.0010 & .0010 & -.0010 & -.2070 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .0110 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

B Matrix for Five Control Surfaces

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ -.5220 & -.3790 & .7140 & .7140 & .0000 \\ -.3950 & -.7740 & -4.8620 & -4.8620 & .0000 \\ .1390 & .2750 & -.1510 & -.1510 & .0000 \\ .0010 & -.0020 & .0000 & .0000 & .0020 \\ .1210 & -.2830 & 1.1590 & -1.5190 & .4310 \\ .0920 & -.1820 & .0030 & -.0030 & -.2070 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \qquad (A.13)$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \qquad (A.14)$$

Auto Regressive Moving Average Models

The coefficients of the autoregressive difference equations begin with Table A.15. The $MATRIX_x$ macro used to generate the matrices is contained in Appendix B.

Table A.15. ACM Entry ARMA Model - No Failures

$$A_1 = \begin{bmatrix} -1.9766D-00 & 0.0000D-00 & 2.0200D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.9877D-00 & 0.0000D-00 & -7.0718D-06 & -8.7077D-05 \\ 3.0638D-03 & 0.0000D-00 & -1.9857D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -27.8260D-00 & 0.0000D-00 & -9.7496D-01 & -7.0019D-01 \\ 0.0000D-00 & 5.7549D-01 & 0.0000D-00 & -6.0038D-04 & -9.7326D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 9.7666D-01 & 0.0000D-00 & -2.0015D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 9.8983D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ -3.0627D-03 & 0.0000D-00 & 9.8814D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 27.7994D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -6.3593D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 7.3484D-04 & 7.3484D-04 & 5.8075D-03 & 5.8075D-03 & 0.0000D-00 \\ -6.7897D-06 & 6.7897D-06 & -6.3897D-06 & 6.3897D-06 & 3.3130D-05 \\ 3.1534D-05 & 3.1534D-05 & -2.1788D-05 & -2.1788D-05 & 0.0000D-00 \\ 2.4002D-05 & -2.4002D-05 & 1.6415D-04 & -1.6415D-04 & 3.4422D-05 \\ 1.2078D-03 & -1.2078D-03 & 1.7236D-04 & -1.7236D-04 & -1.3526D-03 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} -7.7184D-04 & -7.7184D-04 & -5.5416D-03 & -5.5416D-03 & 0.0000D-00 \\ -2.1695D-05 & 2.1695D-05 & 1.3484D-05 & -1.3484D-05 & 3.2708D-06 \\ 3.3706D-05 & 3.3706D-05 & -4.1380D-06 & -4.1380D-06 & 0.0000D-00 \\ -6.0930D-04 & 6.0930D-04 & 3.7870D-04 & -3.7870D-04 & 9.1860D-05 \\ 1.3938D-05 & -1.3938D-05 & -8.6630D-06 & 8.6630D-06 & -2.1013D-06 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table A.16. ACM Entry ARMA Model - 30 Percent Loss of Effectiveness TEL

$$A_1 = \begin{bmatrix} -1.9278D-00 & 0.0000D-00 & 7.2958D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.9879D-00 & 0.0000D-00 & -1.4482D-05 & 1.1316D-04 \\ -1.3075D-02 & 0.0000D-00 & -2.0334D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -21.4570D-00 & 0.0000D-00 & -9.8068D-01 & -5.4042D-01 \\ 0.0000D-00 & 5.6254D-01 & 0.0000D-00 & -5.9587D-04 & -9.7238D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 9.2777D-01 & 0.0000D-00 & -6.7154D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 9.8145D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 1.3073D-02 & 0.0000D-00 & 1.0229D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 21.4414D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -6.2829D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 8.4875D-04 & 8.4975D-04 & 2.9229D-03 & 3.9602D-03 & 0.0000D-00 \\ -1.4104D-05 & 1.3849D-05 & 2.4955D-06 & -4.3548D-06 & 9.4794D-06 \\ 3.1696D-05 & 3.1696D-05 & -1.2768D-05 & -2.1488D-05 & 0.0000D-00 \\ 2.5335D-05 & -2.4091D-05 & 9.5873D-05 & -1.6409D-04 & 3.6092D-05 \\ 1.2172D-03 & -1.1924D-03 & 1.4627D-04 & -2.4366D-04 & 1.3668D-03 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} -1.0088D-03 & -1.0088D-03 & -2.6349D-03 & -3.5456D-03 & 0.0000D-00 \\ -1.3970D-05 & 1.3718D-05 & 2.4856D-06 & -4.3371D-06 & -3.9887D-05 \\ 1.8899D-05 & 1.8899D-05 & -5.0412D-05 & -7.2305D-05 & 0.0000D-00 \\ -3.0520D-04 & 2.9969D-04 & 5.4302D-05 & -9.4750D-05 & -8.7139D-04 \\ 8.9433D-06 & -8.7818D-06 & -1.5912D-06 & 2.7765D-06 & 2.5534D-05 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table A.17. ACM Entry ARMA Model - 50 Percent Loss of Effectiveness CL

$$A_1 = \begin{bmatrix} -1.9488D-00 & 0.0000D-00 & 5.3735D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.9845D-00 & 0.0000D-00 & -1.0175D-05 & -1.0880D-05 \\ -4.4830D-03 & 0.0000D-00 & -2.0113D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -24.8440D-00 & 0.0000D-00 & -9.7764D-01 & -6.2532D-01 \\ 0.0000D-00 & 5.7566D-01 & 0.0000D-00 & -6.0059D-04 & -9.7200D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 9.4885D-01 & 0.0000D-00 & -4.9634D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 9.8666D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 4.4820D-03 & 0.0000D+00 & 1.0077D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 24.8260D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -6.3624D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -4.0629D-03 & -3.1970D-03 & 5.9859D-03 & 5.9859D-03 & 0.0000D-00 \\ -7.0321D-06 & -1.0579D-05 & 3.9925D-06 & -3.9925D-06 & 4.3437D-05 \\ 1.1191D-05 & 2.2381D-05 & -2.1786D-05 & -2.1786D-05 & 0.0000D-00 \\ 1.1082D-05 & -2.1809D-05 & 1.6413D-04 & -1.6413D-04 & 3.4196D-05 \\ 5.9619D-04 & -1.2180D-03 & 1.9454D-04 & -1.9454D-04 & -1.3671D-03 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 3.7323D-03 & 2.7869D-03 & -5.4734D-03 & -5.4734D-03 & 0.0000D-00 \\ -7.0028D-06 & 3.9010D-05 & 3.9740D-06 & -3.9740D-06 & -6.2894D-06 \\ 2.9058D-05 & 3.6021D-05 & -4.8049D-05 & -4.8049D-05 & 0.0000D-00 \\ -1.7621D-04 & 9.8158D-04 & 9.9993D-05 & -9.9993D-05 & -1.5826D-04 \\ 4.5157D-06 & -2.5156D-05 & -2.5626D-06 & 2.5626D-6 & 4.0557D-06 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table A.18. ACM Exit ARMA Model - No Failures

$$A_1 = \begin{bmatrix} -1.9837D-00 & -6.5693D-01 & 2.2386D-00 & 1.1610D-03 & -1.8660D-02 \\ 2.6268D-07 & -1.9658D-00 & -1.1845D-03 & -2.5388D-05 & 4.3827D-04 \\ 4.4540D-04 & 1.4892D-04 & -1.9772D-00 & -1.3834D-04 & 4.2016D-06 \\ 9.3315D-08 & -2.0952D-00 & 1.2440D-03 & -9.9831D-01 & -6.0140D-02 \\ -5.2666D-09 & 1.1718D-01 & -6.6512D-05 & -9.1823D-05 & -9.9858D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 9.8377D-01 & 6.5725D-01 & -2.2302D-00 & 0.0000D-00 & 0.0000D-00 \\ -2.6062D-07 & 9.6796D-01 & 1.1476D-03 & 0.0000D-00 & 0.0000D-00 \\ -4.4191D-04 & -1.4888D-04 & 9.7758D-01 & 0.0000D-00 & 0.0000D-00 \\ -9.2583D-08 & 2.0986D-00 & 1.2332D-03 & 0.0000D-00 & 0.0000D-00 \\ 5.2253D-09 & -1.1514D-01 & -6.8092D-05 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -2.4336D-03 & -2.4560D-03 & -2.3306D-03 & -2.3347D-03 & 6.6196D-07 \\ 1.5424D-05 & -1.8937D-05 & 2.0378D-06 & -2.0392D-06 & 1.7525D-05 \\ 5.0031D-06 & 4.8478D-06 & -1.8059D-06 & -1.8059D-06 & 1.5819D-13 \\ 5.6635D-06 & -1.2379D-05 & 1.9698D-05 & -1.9698D-05 & 6.4000D-06 \\ 3.3248D-04 & -3.0209D-04 & -3.3599D-05 & 3.3599D-05 & -3.2566D-04 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 2.2530D-03 & 2.3031D-03 & 2.4219D-03 & 2.4033D-03 & -2.2257D-06 \\ -1.8931D-05 & 1.5462D-05 & 1.6683D-05 & -1.6687D-05 & -2.2764D-06 \\ 3.9268D-06 & 3.7560D-06 & -2.8621D-06 & -2.8588D-06 & 6.5541D-10 \\ -4.1049D-05 & 3.3518D-05 & 3.6174D-05 & -3.6176D-05 & -4.9354D-06 \\ 2.2522D-06 & -1.8390D-06 & -1.9847D-06 & 1.9848D-06 & 2.7078D-07 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table A.19. TF/TA ARMA Model - No Failures

$$A_1 = \begin{bmatrix} -3.6847D-00 & 0.0000D-00 & -51.1370D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.9519D-00 & 0.0000D-00 & -2.8733D-05 & 1.0815D-04 \\ 5.9207D-02 & 0.0000D-00 & -1.8445D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -59.2300D-00 & 0.0000D-00 & -9.5265D-01 & -1.5044D-00 \\ 0.0000D-00 & 1.4404D-00 & 0.0000D-00 & -1.3954D-03 & -9.2447D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 2.6832D-00 & 0.0000D-00 & 49.7810D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 9.5785D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ -5.9154D-02 & 0.0000D-00 & -7.6791D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 59.0480D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.6383D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 1.0618D-02 & -1.0618D-02 & 1.5259D-02 & 1.5259D-02 & 0.0000D-00 \\ -1.8033D-05 & 1.8033D-05 & -4.1580D-06 & 9.1373D-06 & 1.0194D-04 \\ 7.3451D-05 & 7.3451D-05 & -4.0607D-05 & -4.0607D-05 & 0.0000D-00 \\ 7.1020D-05 & -7.1020D-05 & 2.9993D-04 & -2.9997D-04 & 1.0971D-04 \\ 3.9102D-03 & -3.9102D-03 & 1.0870D-04 & -1.0819D-04 & -4.4117D-03 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 2.0494D-02 & 2.0494D-02 & -3.6496D-02 & -3.6496D-02 & 0.0000D-00 \\ -7.5883D-05 & 7.5883D-05 & 1.0468D-05 & -1.5268D-05 & 1.2956D-05 \\ -4.2691D-04 & -4.2691D-04 & 7.9098D-04 & 7.9098D-04 & 0.0000D-00 \\ -4.6779D-03 & 4.6779D-03 & 6.4531D-04 & -9.4120D-04 & 7.9867D-04 \\ 1.2979D-04 & -1.2979D-04 & -1.7904D-05 & 2.6114D-05 & -2.2160D-05 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table A.20. TF/TA ARMA Model - 30 Percent Loss of Effectiveness TEL

$$A_1 = \begin{bmatrix} -1.9289D-00 & 0.0000D-00 & 2.4122D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.9561D-00 & 0.0000D-00 & -2.5617D-05 & 1.3755D-05 \\ 7.2371D-03 & 0.0000D-00 & -1.9644D-00 & 0.0000D+00 & 0.0000D-00 \\ 0.0000D-00 & 57.5160D-00 & 0.0000D-00 & -9.5402D-01 & -1.4610D-00 \\ 0.0000D-00 & 1.6261D-00 & 0.0000D-00 & -1.5406D-03 & -9.2005D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 9.2900D-01 & 0.0000D-00 & -2.3558D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 9.6193D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ -7.2311D-03 & 0.0000D-00 & 9.7021D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 57.7350D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.8209D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -8.8814D-03 & -8.8814D-03 & 1.0201D-02 & 1.3521D-02 & 0.0000D-00 \\ -2.2384D-05 & 2.2384D-05 & 2.2204D-06 & -3.6512D-06 & 1.0626D-04 \\ 7.2684D-05 & 7.2684D-05 & -2.3547D-05 & -3.9662D-05 & 0.0000D-00 \\ 7.0004D-05 & -7.0004D-05 & 1.7183D-04 & -2.9585D-04 & 1.1159D-04 \\ 3.8607D-03 & -3.8607D-03 & 4.3589D-05 & -8.8733D-05 & -4.3508D-03 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 8.2180D-03 & 8.2180D-03 & -9.3663D-03 & -1.2370D-02 & 0.0000D-00 \\ -7.0757D-05 & 7.0757D-05 & 2.1920D-06 & -3.6047D-06 & 7.5178D-06 \\ 7.4294D-06 & 7.4294D-06 & 5.0055D-05 & 5.7807D-05 & 0.0000D-00 \\ -4.2185D-03 & 4.2185D-03 & 1.3069D-04 & -2.1491D-04 & 4.4821D-04 \\ 1.3394D-04 & -1.3394D-04 & -4.1495D-06 & 6.8238D-06 & -1.4231D-05 \end{bmatrix}$$

$$\theta^T = [A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}]$$

Table A.21. TF/TA ARMA Model - 50 Percent Loss of Effectiveness CL

$$A_1 = \begin{bmatrix} -1.9180D-00 & 0.0000D-00 & 2.4247D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -1.9713D-00 & 0.0000D-00 & -1.2614D-05 & -5.2084D-04 \\ 4.7557D-03 & 0.0000D-00 & -1.9509D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -83.6540D-00 & 0.0000D-00 & -9.3315D-01 & -2.1289D-00 \\ 0.0000D-00 & 2.0604D-00 & 0.0000D-00 & -1.9038D-03 & -9.0350D-01 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 9.1806D-01 & 0.0000D-00 & -2.3596D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 9.7736D-01 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ -4.7512D-03 & 0.0000D-00 & 9.5472D-01 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & 83.3500D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \\ 0.0000D-00 & -2.2661D-00 & 0.0000D-00 & 0.0000D-00 & 0.0000D-00 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -1.3180D-02 & -9.7372D-03 & 1.6645D-02 & 1.6645D-02 & 0.0000D-00 \\ -3.0171D-06 & 5.2788D-06 & 3.0283D-06 & -3.0283D-06 & 1.1496D-04 \\ 4.2701D-05 & 8.4482D-05 & -4.6574D-05 & -4.6574D-05 & 0.0000D-00 \\ 3.6503D-05 & -8.5166D-05 & 3.4423D-04 & -3.4423D-04 & 1.2590D-04 \\ 2.2580D-03 & -4.4667D-03 & 6.4002D-05 & -6.4002D-05 & -5.0749D-03 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 1.2025D-02 & 8.7762D-03 & -1.4993D-02 & -1.4993D-02 & 0.0000D-00 \\ -5.2586D-05 & 1.0441D-04 & 2.9923D-06 & -2.9923D-06 & 1.6596D-05 \\ -2.0516D-05 & 3.7108D-05 & 3.2466D-05 & 3.2466D-05 & 0.0000D-00 \\ -4.4845D-03 & 8.9044D-03 & 2.5519D-04 & -2.5519D-04 & 1.4153D-03 \\ 1.2193D-04 & -2.4210D-04 & -6.9381D-06 & 6.9381D-06 & -3.8481D-05 \end{bmatrix}$$

$$\theta^T = A_{11}^{(1)} A_{12}^{(1)} \ldots A_{55}^{(1)} A_{11}^{(2)} \ldots A_{55}^{(2)} B_{11}^{(1)} \ldots B_{55}^{(1)} B_{11}^{(2)} \ldots B_{55}^{(2)}$$

Appendix B. *ARMA Model Implementation*

Introduction

This appendix contains a summary of the ARMA model conversion technique developed by Bokor and Keviczky and the MATRIX_x macro used to accomplish this transformation for all the plant matrices [5,25]. Detailed mathematical development of the procedure is contained in reference [5].

ARMA Model

The method used to generate the ARMA model representation from a state-space model representation is based upon using constructibility invariants. This method eliminates some of the problems associated with the observability matrix technique when the state transition matrix is singular. Recall that the state-space representation of the continuous time system is expressed by Equation 2.1 and Equation 2.2

$$\dot{x} = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

where,

A = the continuous time plant matrix (n x n)

B = the continuous time control matrix (n x m) with the rank or B = m

C = the continuous time output matrix (p x n)

x = the state variable vector with n states

u = the control input vector with m inputs

y = the output vector with p outputs

The state-space matrices of Equation 2.1 and Equation 2.2 are discretized for a given sampling time T and expressed in the discrete time domain by

$$\dot{x}(T) = Fx(T) + Gu(T) \tag{B.1}$$

$$y(T) = Hx(T) \tag{B.2}$$

where,

F = the discrete plant matrix (n x n)

G = the discrete control matrix (n x m) with the rank = m

H = the discrete output matrix (p x n)

x = the state variable vector with n states

u = the control input vector with m inputs

y = the output vector with p outputs

For this technique, the discrete time representation Equations B.1 and B.2 must have no poles at the origin, i.e. the inverse matrix F^{-1} exists. The constructibility matrix C_o is defined as

$$C_o = \begin{bmatrix} HF^{-1} \\ HF^{-2} \\ HF^{-3} \\ \vdots \\ HF^{-n} \end{bmatrix} \tag{B.3}$$

and must have rank $C_0 = n$.

Equation B.3 is expanded to yield the form

$$C_o = \begin{bmatrix} h_1^T F^{-1} & \dots & h_p^T F^{-1} \\ h_1^T F^{-2} & \dots & h_p^T F^{-2} \\ h_1^T F^{-3} & \dots & h_p^T F^{-3} \\ & \dots & \end{bmatrix} \tag{B.4}$$

A new matrix, T, is formed from the linearly independent rows of C_o. The first row of the new matrix is the first row of $C_o(H_1 F^{-1})$. The second row is the next linearly independent row of C_o. For instance, if $C_o(H_2 F^{-1})$ is linearly independent of $C_o(H_1 F^{-1})$ then it is the second row of the new matrix, T. The process continues until the matrix T has full rank n.

The transformation matrix TF is formed by rearranging the rows of matrix T into the following form

$$TF = \begin{bmatrix} h_1^T F^{-1} \\ \vdots \\ h_1^T F^{-V_1} \\ \hline \vdots \\ \hline h_p^T F^{-1} \\ \vdots \\ h_p^T F^{-V_p} \end{bmatrix} \tag{B.5}$$

For $1 \leq i \leq p$, the ith constructibility index V_i is defined as the smallest positive integer such that $h_i^T F^{-V_i-1}$ is a linear combination of the rows before it. Then the constructibility indices satisfy the relation

$$V_1 + V_2 + \cdots + V_p = n \tag{B.6}$$

and are arranged in descending order

$$V_1 \geq V_2 \geq V_3 > \cdots V_p \tag{B.7}$$

If Equation B.7 is *not* satisfied, permutate the output matrix, H, to satisfy the relationship of the constructibility indices.

Then the following matrices are defined:

$$R = T^{-1} \tag{B.8}$$

$$\bar{H} = HR \tag{B.9}$$

$$\bar{F} = R^{-1}FR \tag{B.10}$$

$$\bar{G} = R^{-1}G \tag{B.11}$$

The A_i coefficients of the ARMA model are calculated from the relationship

$$A_i = \bar{H}Sp_i(k-1) \tag{B.12}$$

where,

$$S_p(k) = \begin{bmatrix} (k-1) & 0 & \dots & 0 \\ (k-2) & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ (k-V_1) & 0 & \dots & 0 \\ 0 & (k-1) & \dots & 0 \\ 0 & (k-2) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & (k-V_2) & \dots & 0 \\ 0 & 0 & \dots & (k-1) \\ 0 & 0 & \dots & (k-2) \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & (k-V_p) \end{bmatrix} (n \text{ x } p) \qquad (B.13)$$

As an example, to find the A_1 coefficients using Equation B.12 and Equation B.13, the $S_p(k)$ matrix elements associated with the (k-1) terms in Equation B.13 would be set equal to 1 and all other matrix elements equal to zero. The A_2 coefficients would require $S_p(k)$ matrix elements associated with the (k-2) terms in Equation B.13 be set equal to 1 and all other matrix elements equal to zero, etc.

The B_i coefficients of the ARMA model are calculated from the relationship

$$B_i = \bar{H} S_q(k-i)\bar{G} \qquad (B.14)$$

where,

$$S_q = \sum_{i=1}^{\bar{v}} S_{i-1} D_i, \bar{v} = \max V_i \qquad (B.15)$$

and S_i is the block-diagonal Toeplitz matrix give by

$$S_i = \begin{bmatrix} 0 & \dots & 0 \\ 1 & \dots & \vdots \\ 0 & 1 & 0 \end{bmatrix} (V_i \text{ x } V_i) \qquad (B.16)$$

and

$$S = \text{ block diagonal } [S_1, \dots, S_p] \qquad (B.17)$$

ARMA Implementation

The implementation of the theory presented in the previous section is executed in MATRIX_x using a modification of a macro developed by Velez in his thesis [25]. For this design,

$$V_1 = 2 \qquad (B.18)$$

$$V_2 = 2 \qquad (B.19)$$

$$V_3 = 2 \qquad (B.20)$$

$$V_4 = 1 \qquad (B.21)$$

$$V_5 = 1 \qquad (B.22)$$

$$S_{p_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (B.23)$$

$$
S_{p_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.24}
$$

$$
S_q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.25}
$$

MATRIX$_x$ ARMA Macro The following MATRIX$_x$ macro listing can be directly typed into MATRIX$_x$ and implemented by using the command

EXEC('filename').

```
S=<A,B;C,O*EYE(5)>;       //FORMS THE S PLANE SYSTEM MATRIX
SD=DISC(S,8,.025);        //DISCRETIZES THE SYSTEM MATRIX
<F,G,H,D>=SPLIT(SD,8);    //SEPARATES DISCRETE MATRICES
CO=<H*INV(F)>;            //FORMS 1ST ROW OF CONSTRUCTIBILITY
FOR I=2:8,...
    CO=<CO;H*INV(F)**I>;...
END;                      //FORMS THE REMAINING ROWS OF CO
V1=O;                     //INITIALIZE INDICES
CR=O;                     //INITIALIZE COUNTER
TT=<CO(1,:)>;             //INITIALIZE TEMPORARY VECTOR 1ST ROW
R=TT;                    //INITIALIZE TEMPORARY VECTOR
```

```
FOR I=2:16,...
     R=<TT;CO(I,:)>;...
     RN=RANK(R);...
     IF RN=I-CR THEN TT=R;...
        ELSEIF RN<I-CR THEN R=TT;...
        VI=<V1;I>;...
        V1=VI;...
        CR=CR+1;...
    END,...
END;
T=R;
TF=<T(1,:);T(6,:);T(2,:);T(7,:);...
    T(3,:);T(8,:);T(4,:);T(5,:)>;
HBAR=H*INV(TF);
FBAR=TF*F*INV(TF);
GBAR=TF*G;
SQ=<0 0 0 0 0 0 0 0;...
    1 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0;...
    0 0 1 0 0 0 0 0;...
    0 0 0 0 0 0 0 0;...
    0 0 0 0 1 0 0 0;...
    0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0>;
SP1=<1 0 0 0 0;...
     0 0 0 0 0;...
     0 1 0 0 0;...
     0 0 0 0 0;...
     0 0 1 0 0;...
     0 0 0 0 0;...
     0 0 0 1 0;...
     0 0 0 0 1>;
SP2=<0 0 0 0 0;...
     1 0 0 0 0;...
     0 0 0 0 0;...
     0 1 0 0 0;...
     0 0 0 0 0;...
     0 0 1 0 0;...
     0 0 0 0 0;...
     0 0 0 0 0>;
B1ARMA=HBAR*GBAR;                       //B1 COEFFICIENT
B2ARMA=HBAR*SQ*GBAR;                    //B2 COEFFICIENT
```

```
A1ARMA=-HBAR*SP1;                         //A1 COEFFICIENT
A2ARMA=-HBAR*SP2;                         //A2 COEFFICIENT
```

Appendix C. *Parameter Vector Elements*

Introduction

Chapter 2 and 3 introduce the parameter estimation scheme used in this thesis and Chapter 6 develops the implementation of the parameter adaptive algorithm. This chapter shows the time history of the changes in the 100 parameter vector elements as the control surface failure is introduced during the simulation. However, the time history of the parameter elements is extended to 15 seconds in order to observe the convergence of the parameter values.

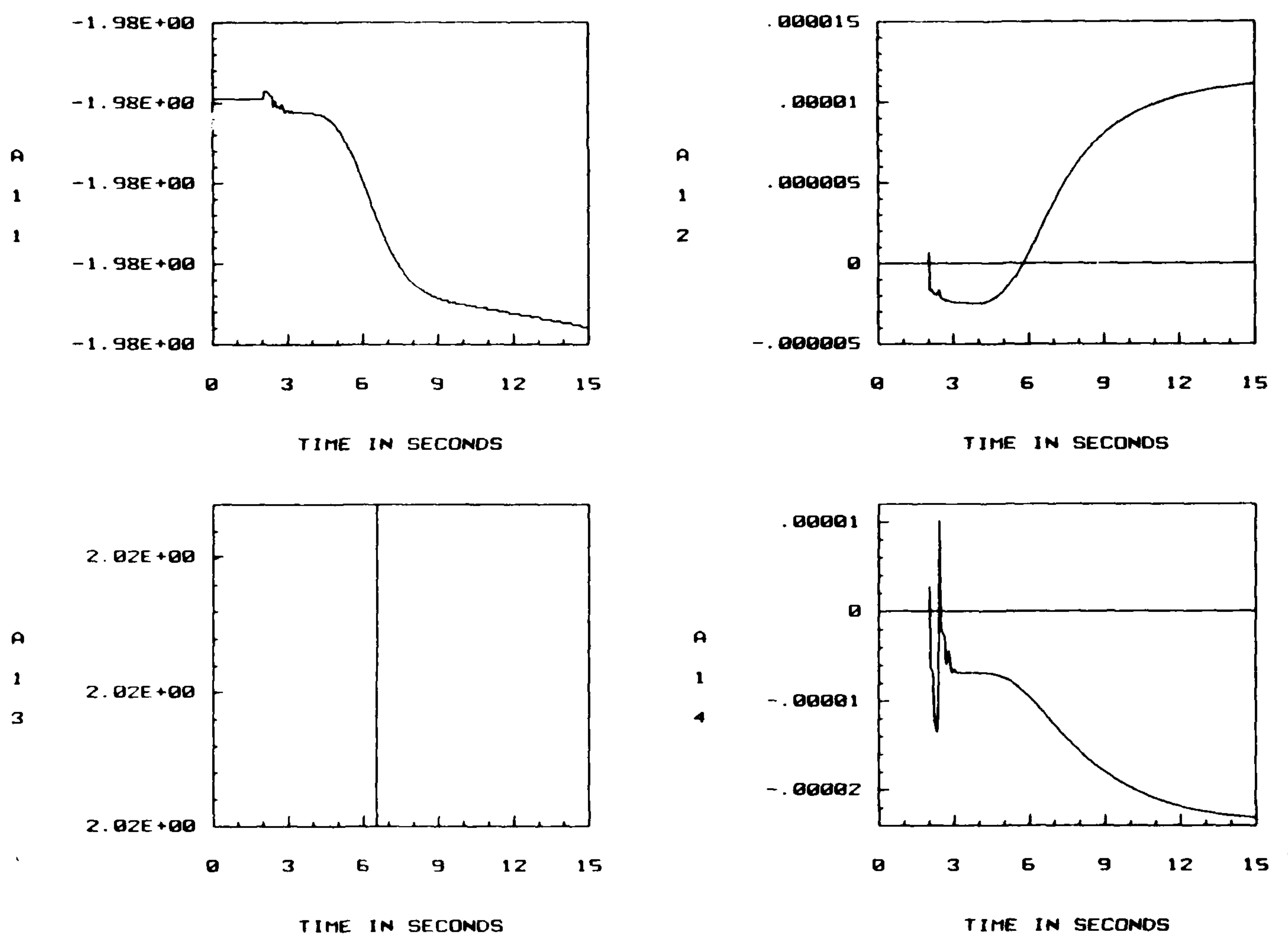


Figure C.1. A_1 ARMA Coefficients - $A_{11}, A_{12}, A_{13}, A_{14}$

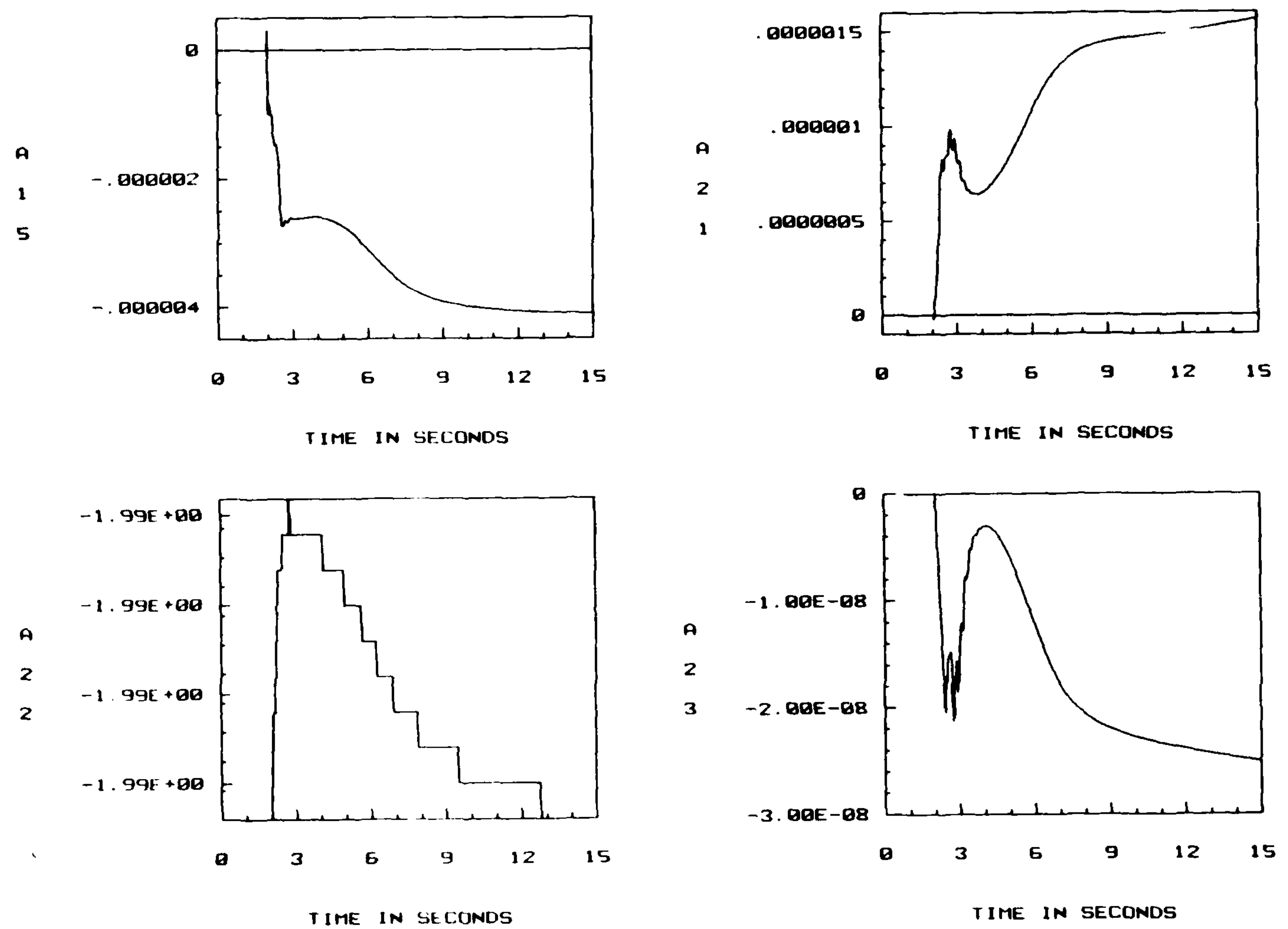


Figure C.2. A_1 ARMA Coefficients - $A_{15}, A_{21}, A_{22}, A_{23}$

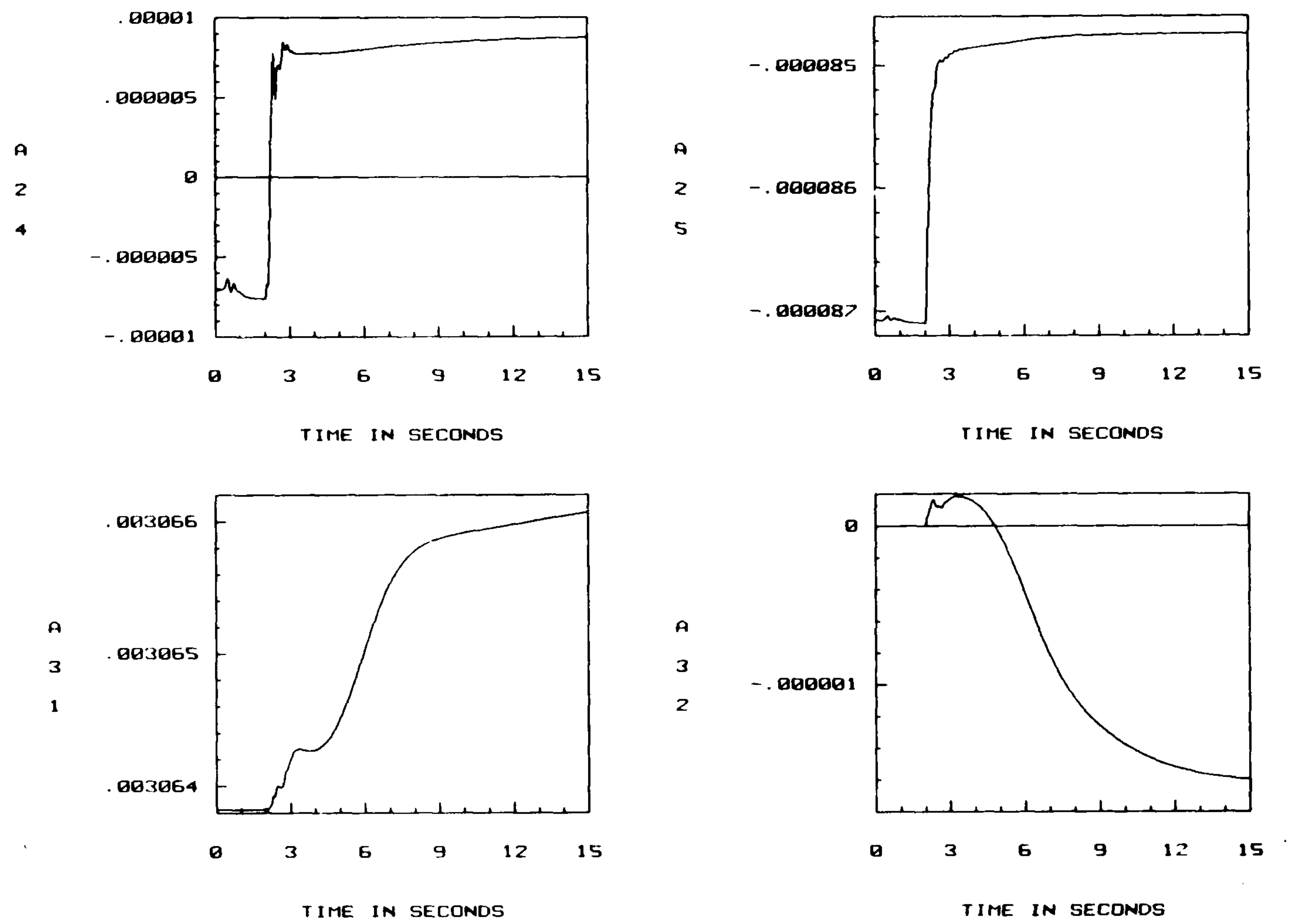


Figure C.3. A_1 ARMA Coefficients - $A_{24}, A_{25}, A_{31}, A_{32}$

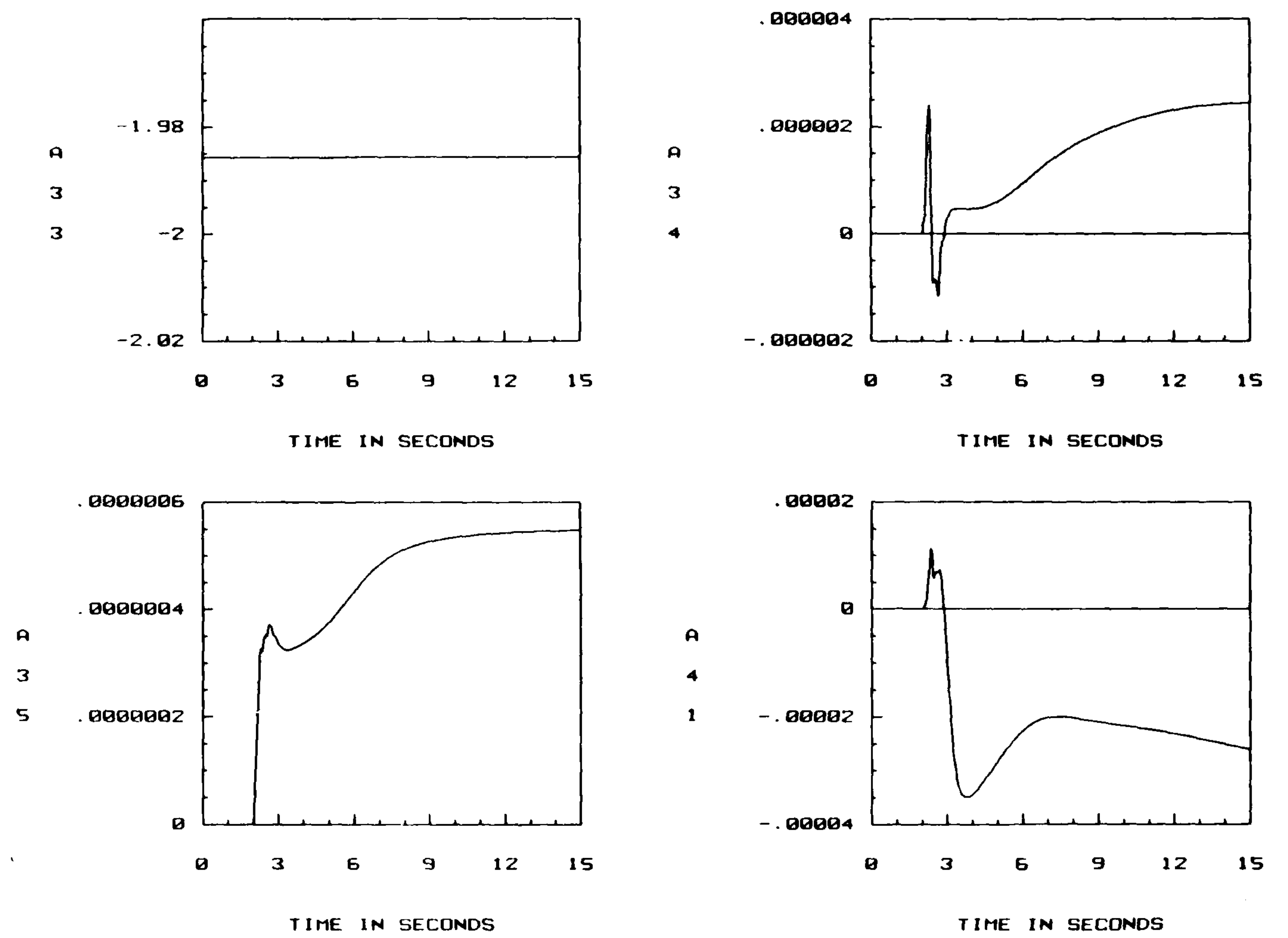


Figure C.4. A_1 ARMA Coefficients - $A_{33}, A_{34}, A_{35}, A_{41}$

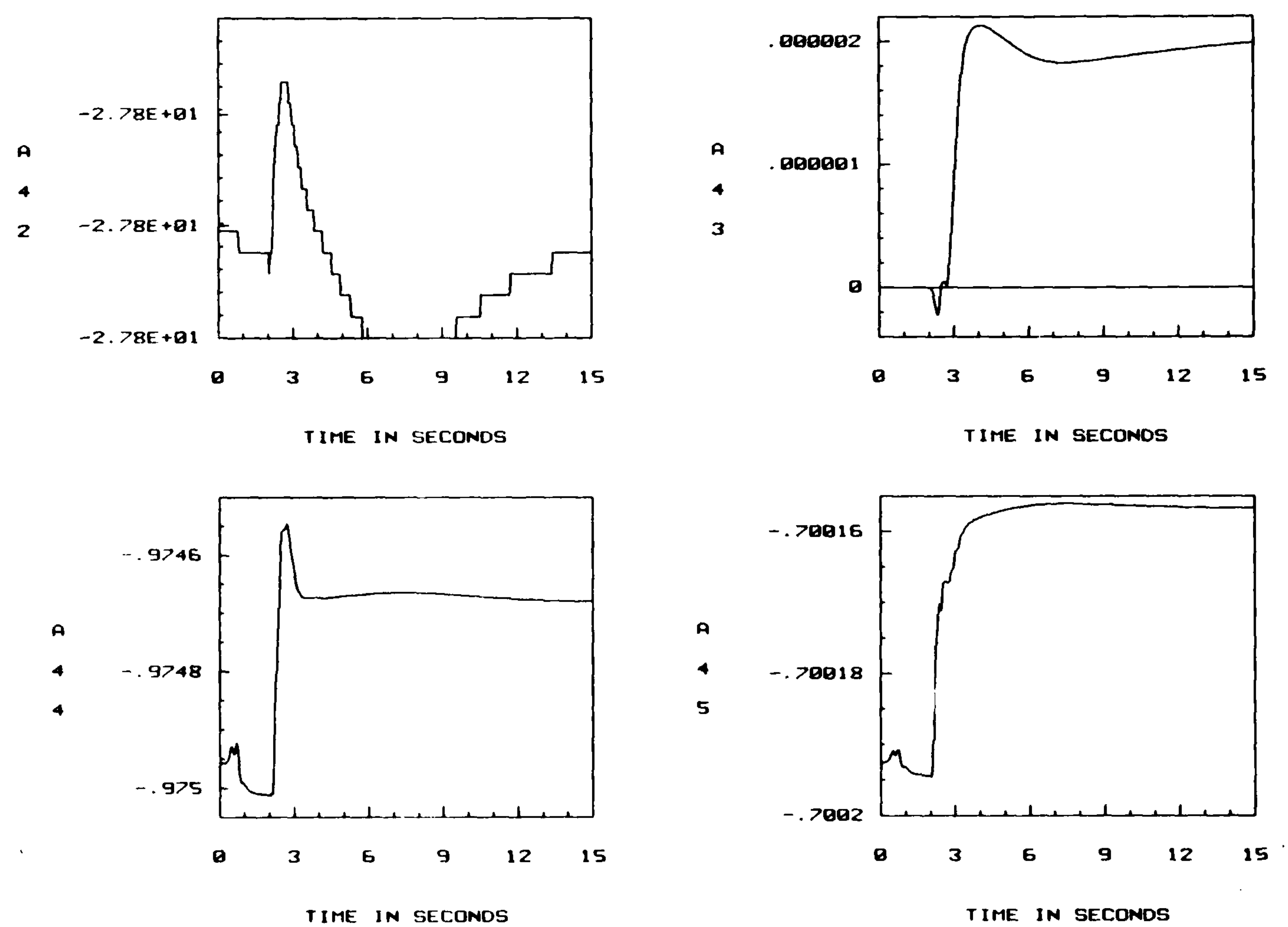


Figure C.5. A_1 ARMA Coefficients - $A_{42}, A_{43}, A_{44}, A_{45}$

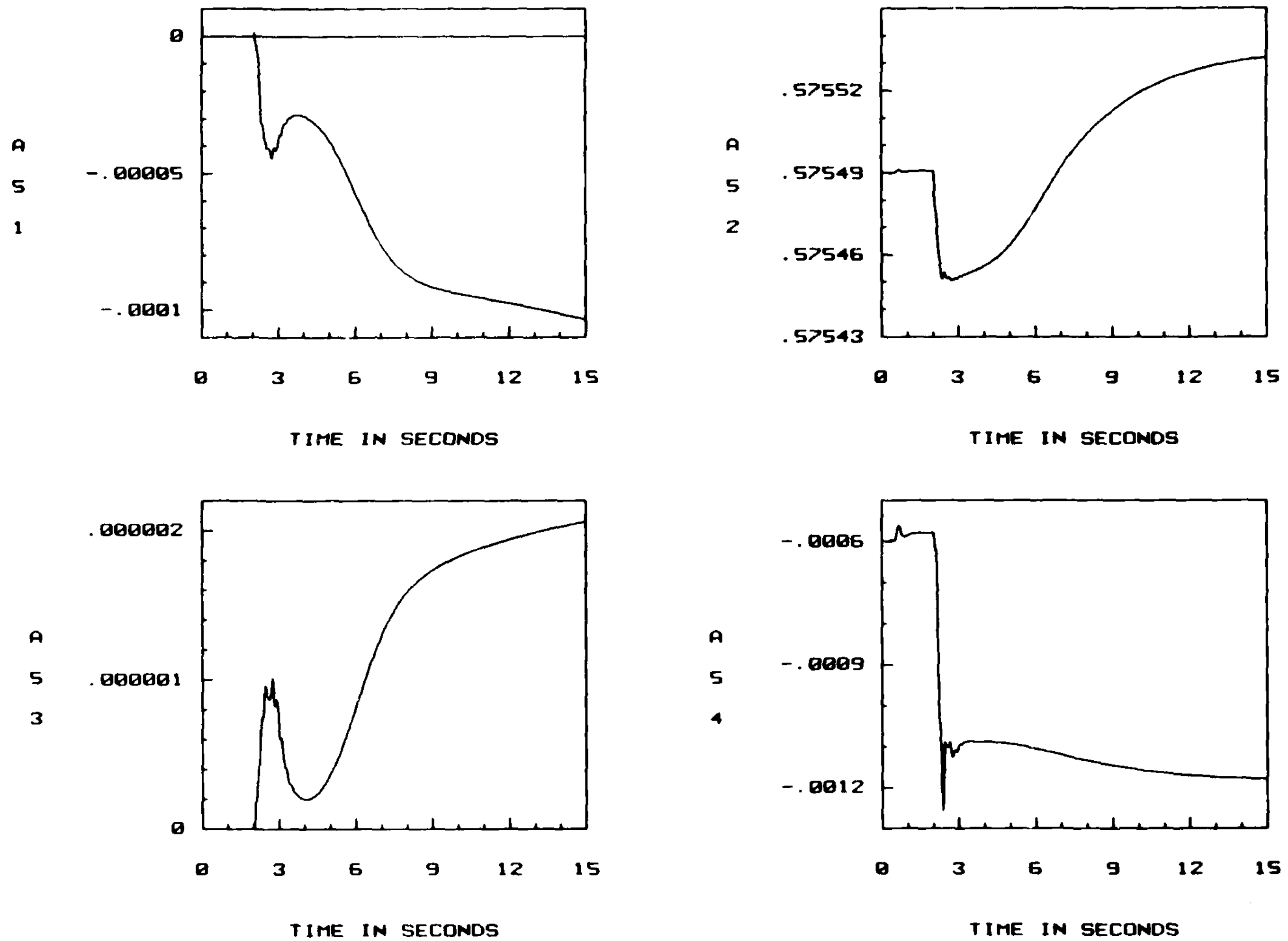


Figure C.6. A_1 ARMA Coefficients - $A_{51}, A_{52}, A_{53}, A_{54}$

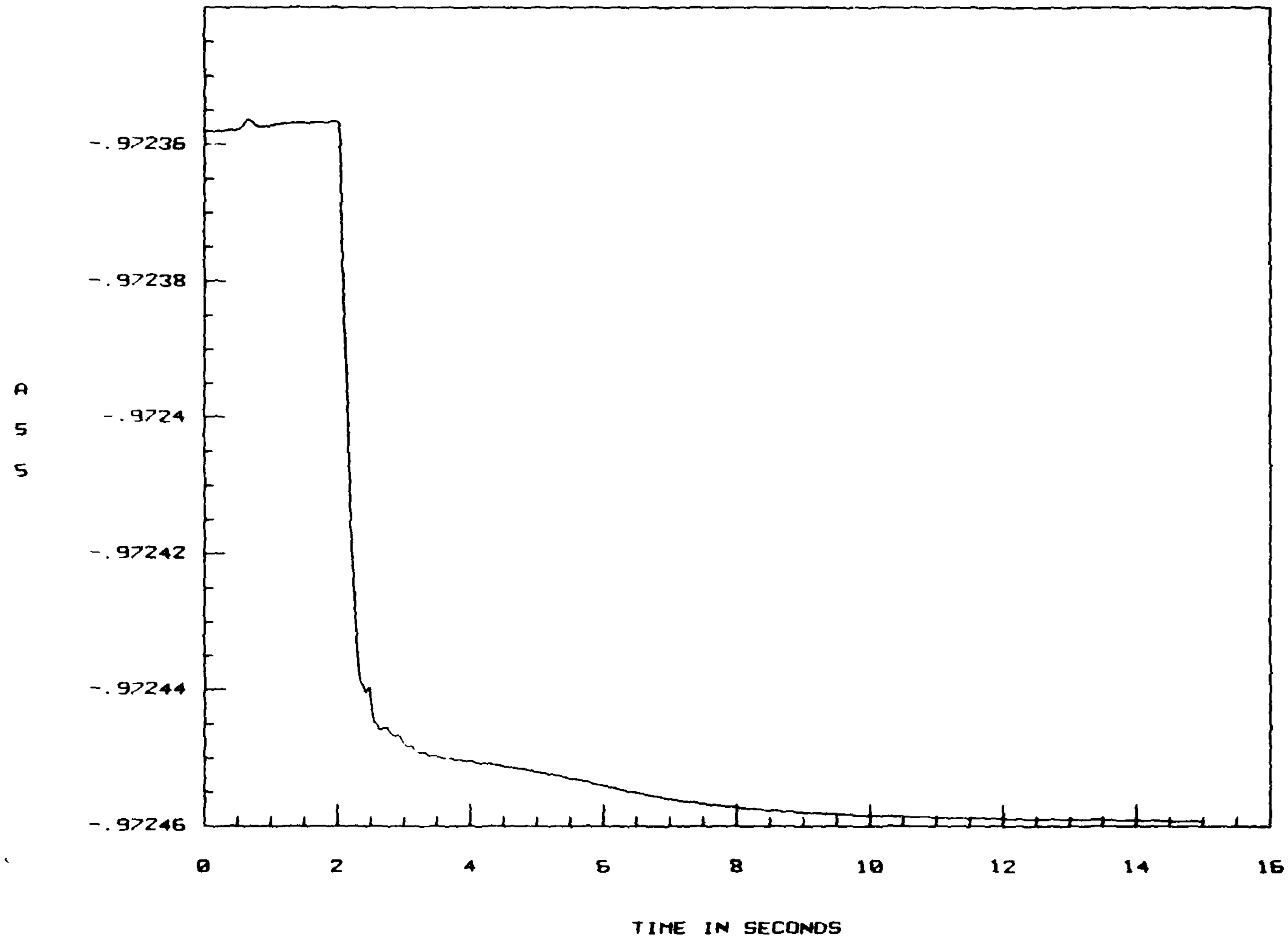


Figure C.7. A_1 ARMA Coefficients - A_{55}

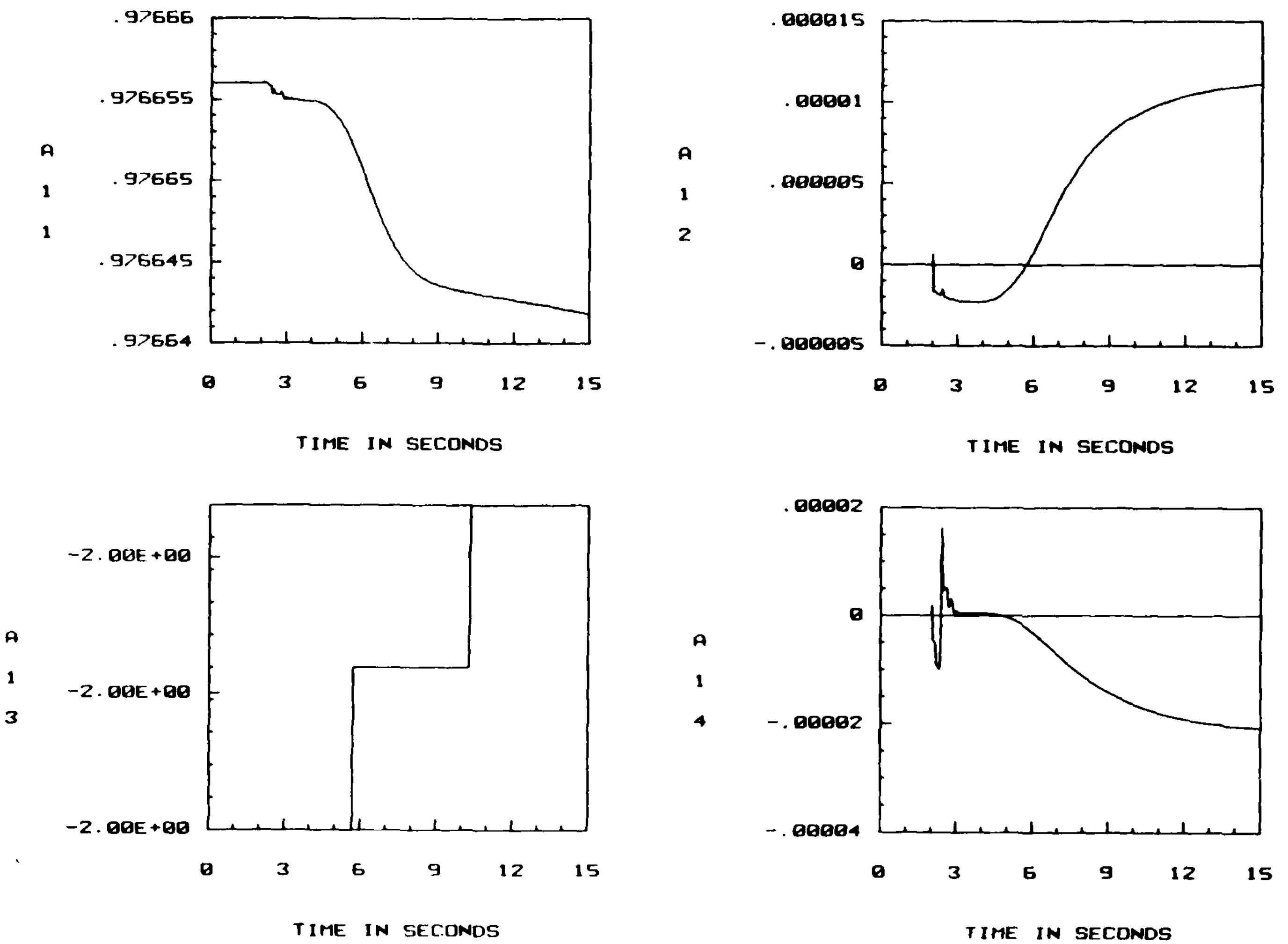


Figure C.8. A_2 ARMA Coefficients - $A_{11}, A_{12}, A_{13}, A_{14}$

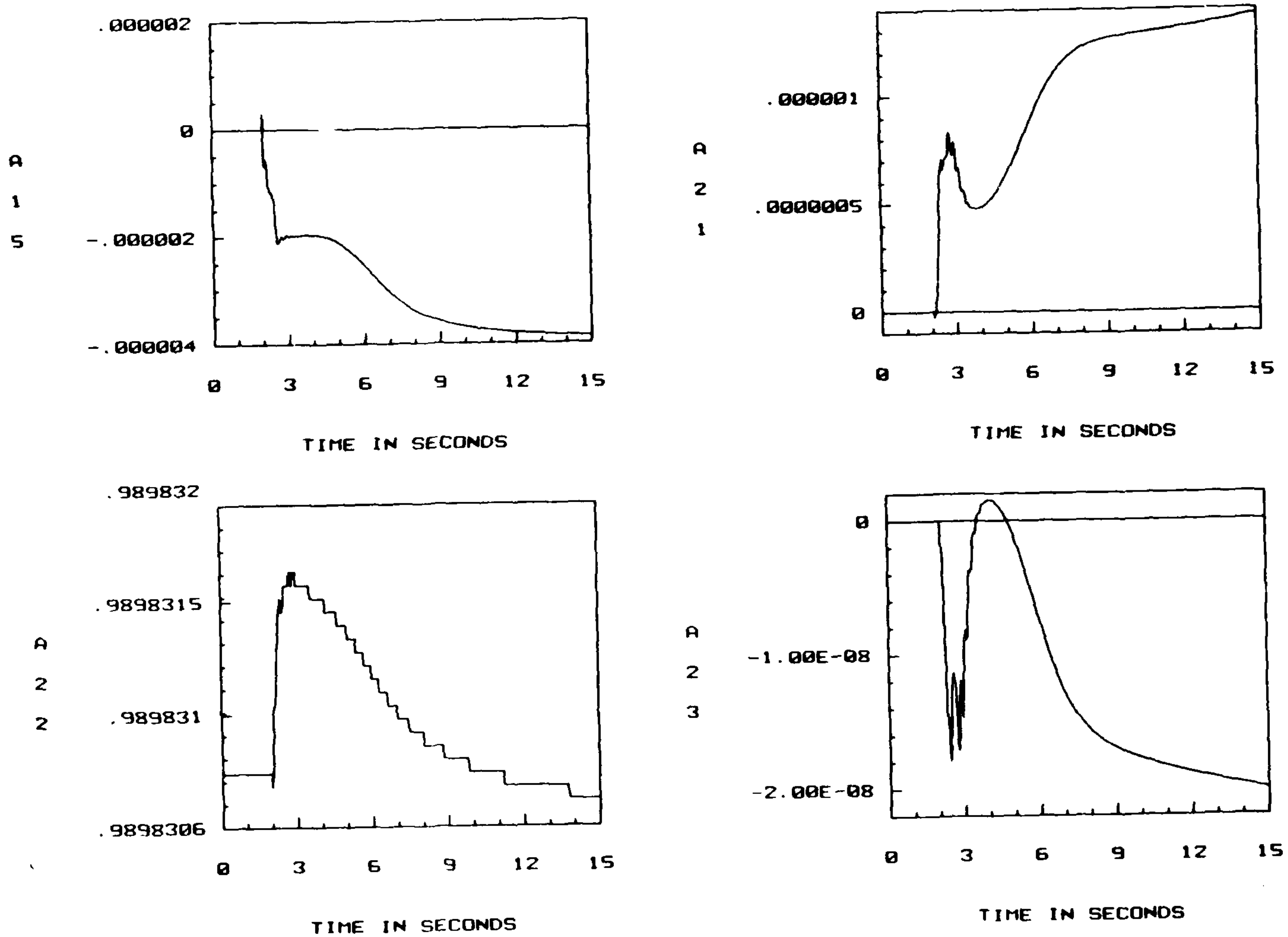


Figure C.9. A_2 ARMA Coefficients - $A_{15}, A_{21}, A_{22}, A_{23}$

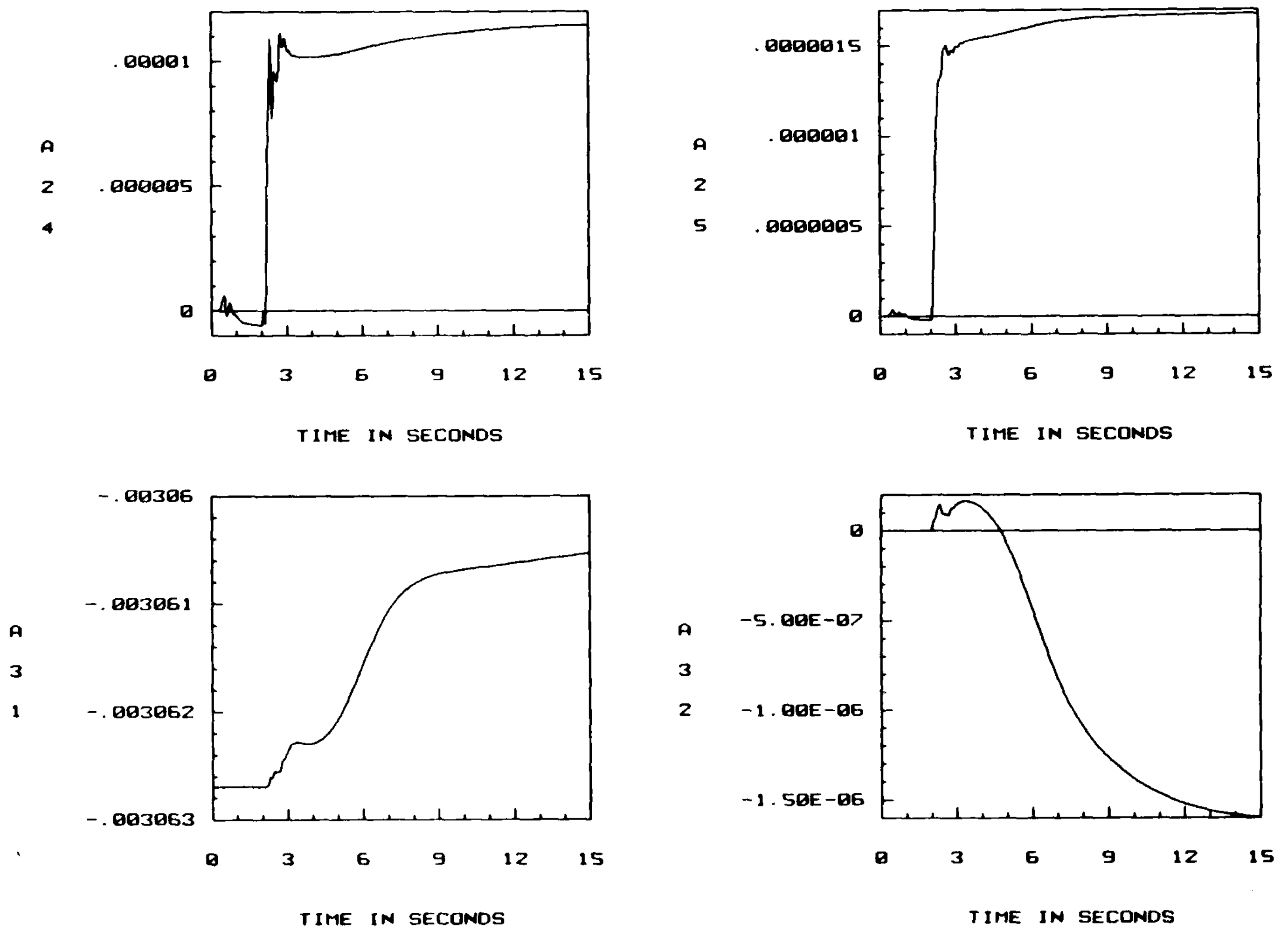


Figure C.10. A_2 ARMA Coefficients - $A_{24}, A_{25}, A_{31}, A_{32}$

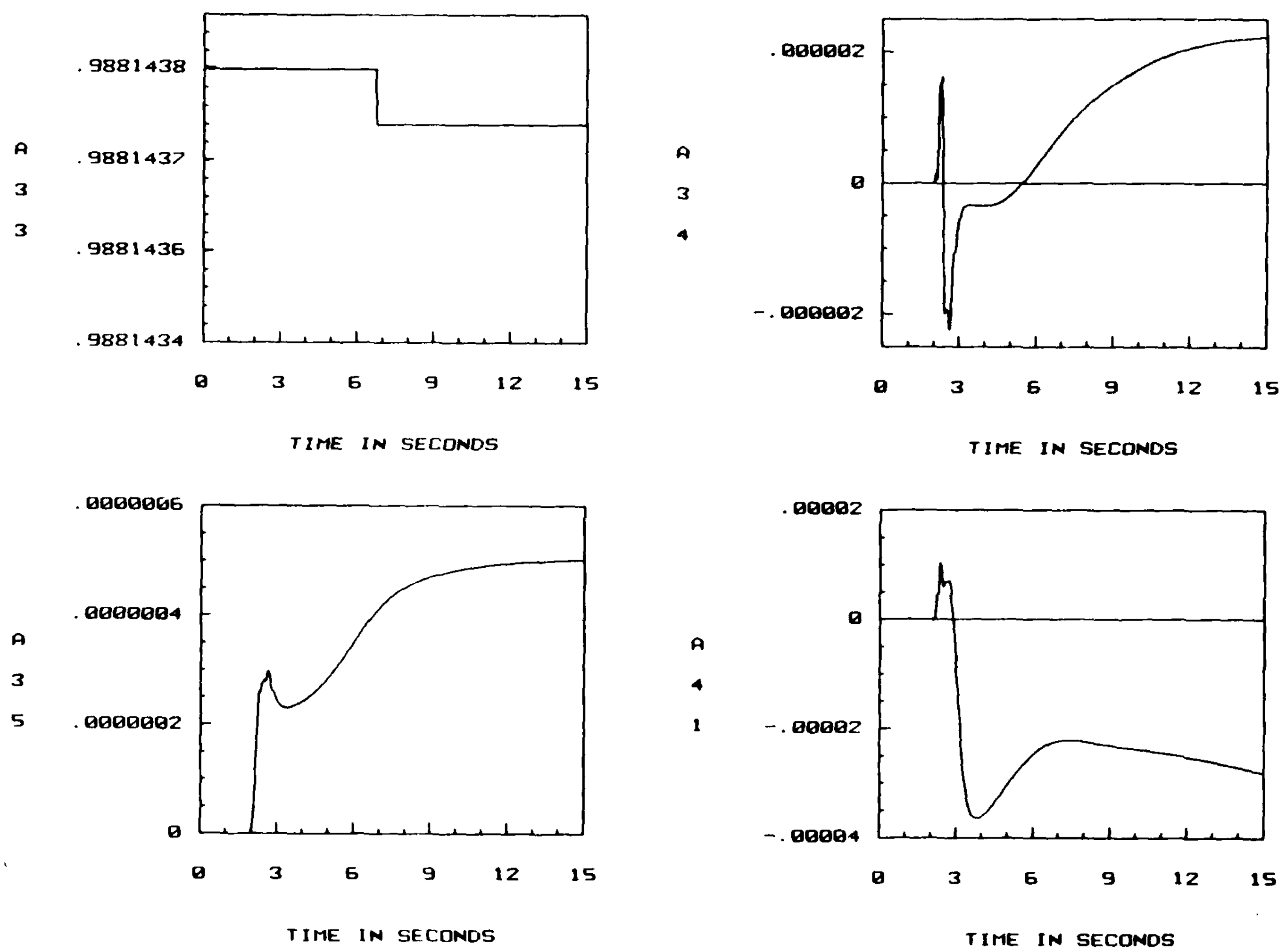


Figure C.11. A_2 ARMA Coefficients - $A_{33}, A_{34}, A_{35}, A_{41}$

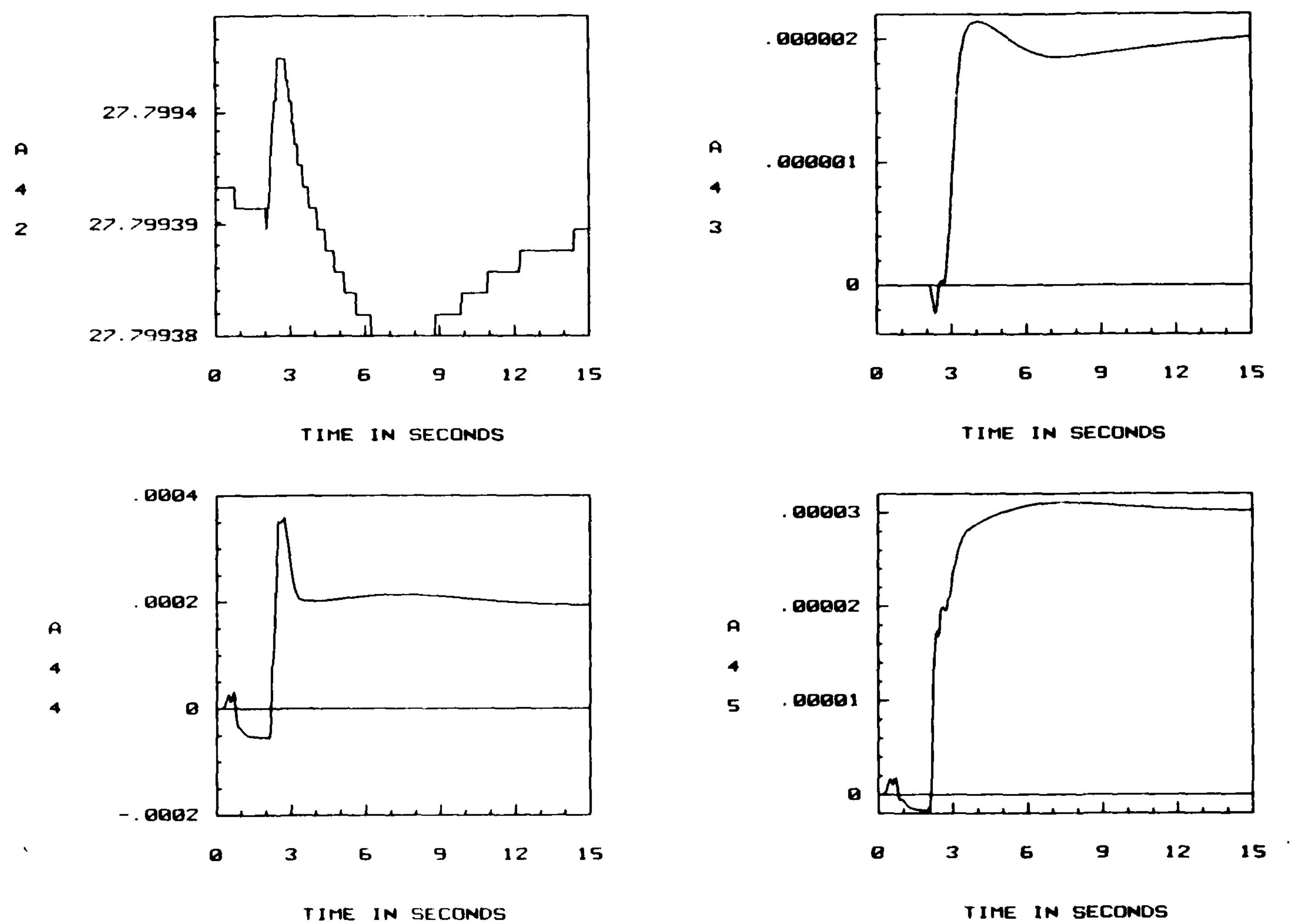


Figure C.12. A_2 ARMA Coefficients - $A_{42}, A_{43}, A_{44}, A_{45}$

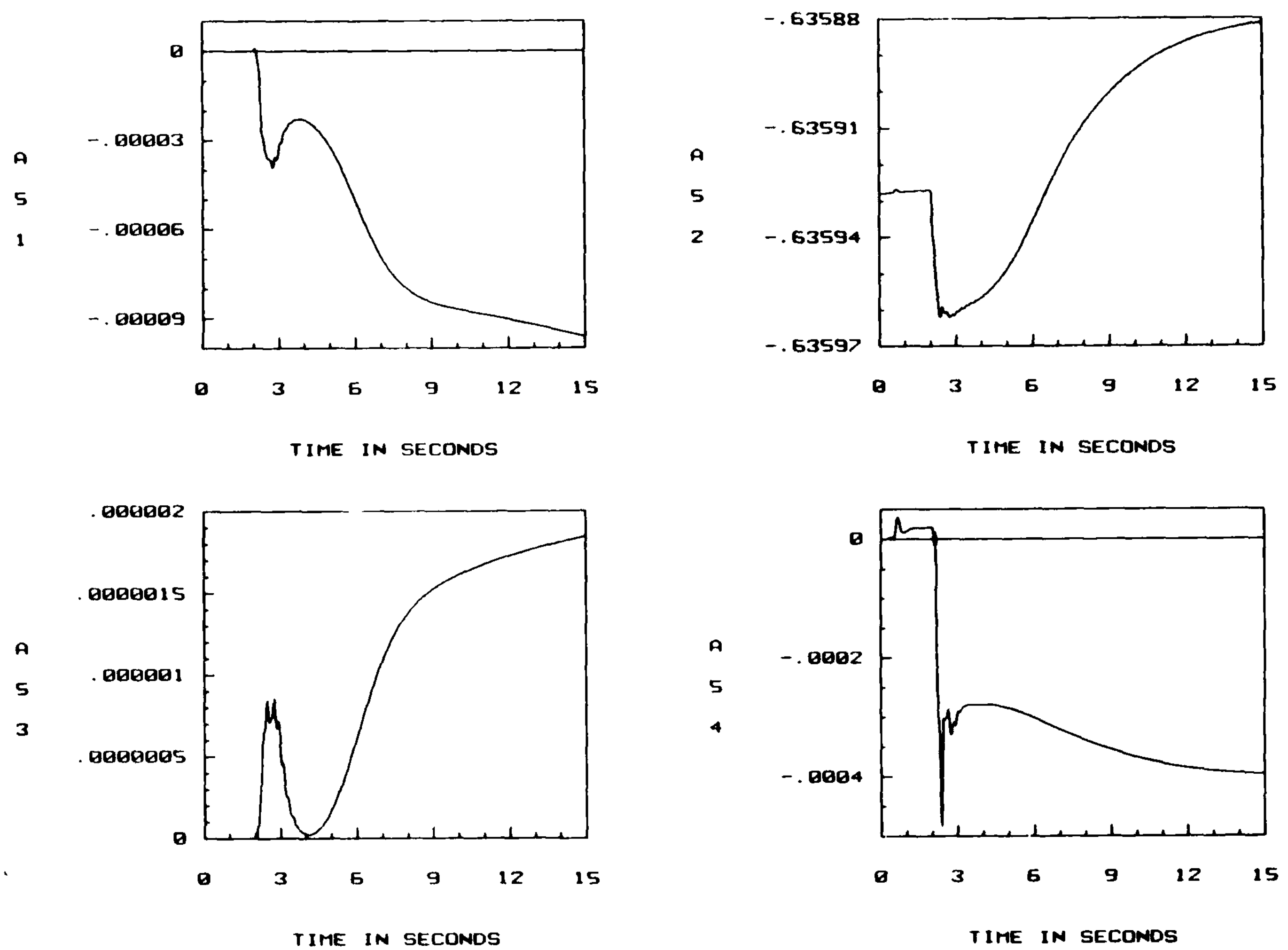


Figure C.13. A_2 ARMA Coefficients - $A_{51}, A_{52}, A_{53}, A_{54}$

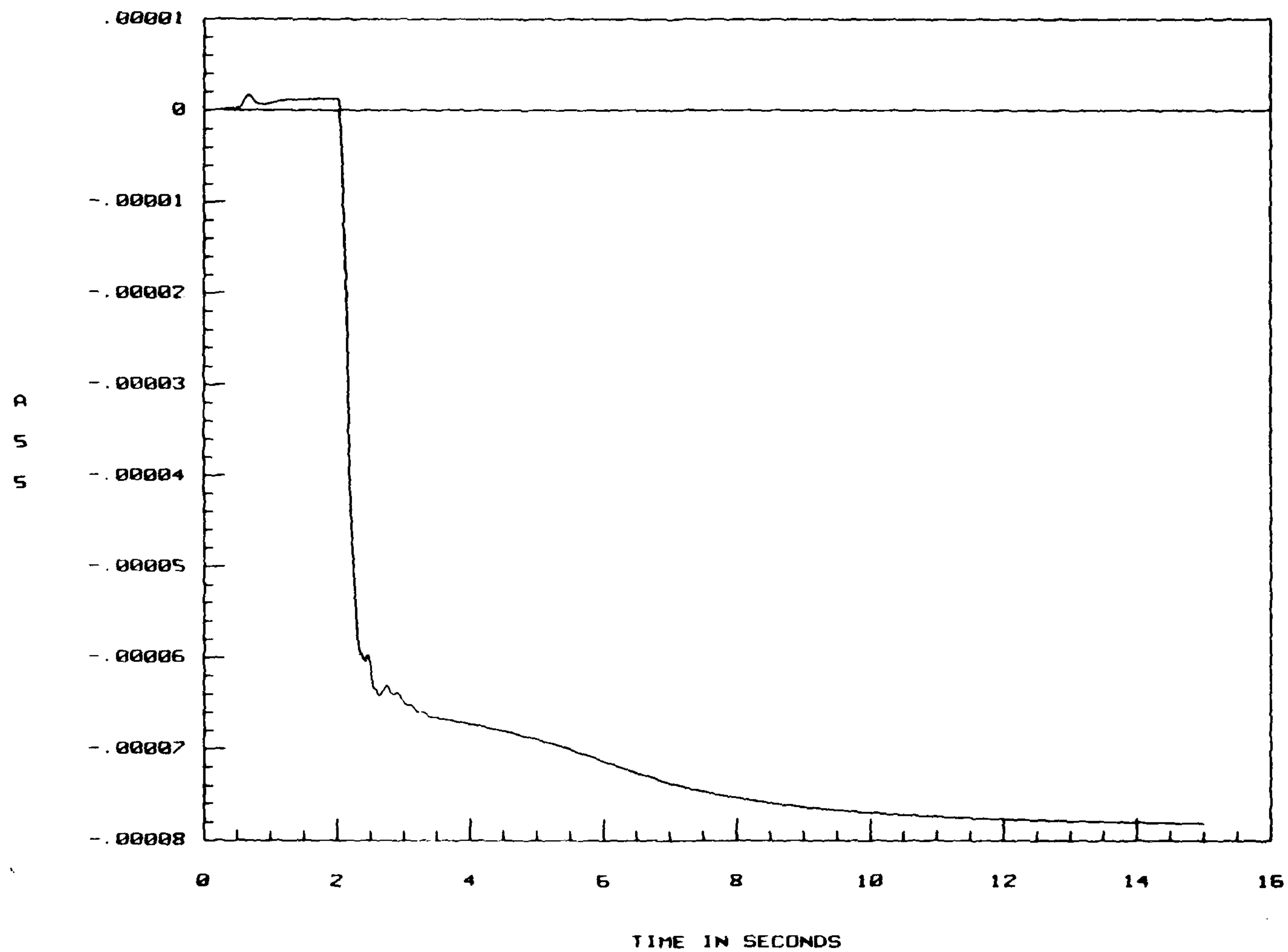


Figure C.14. A_2 ARMA Coefficients - A_{55}

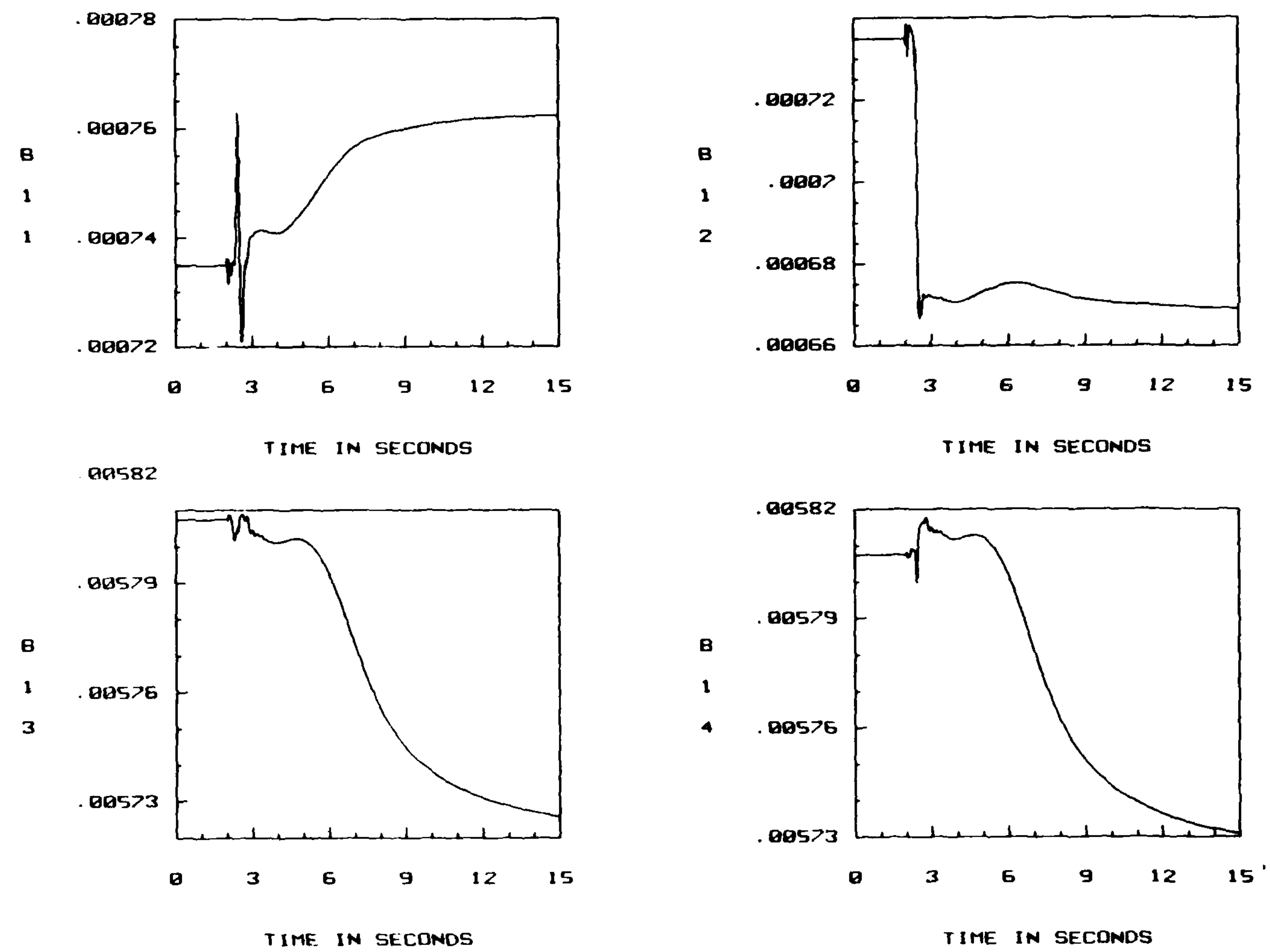


Figure C.15. B_1 ARMA Coefficients - $B_{11}, B_{12}, B_{13}, B_{14}$

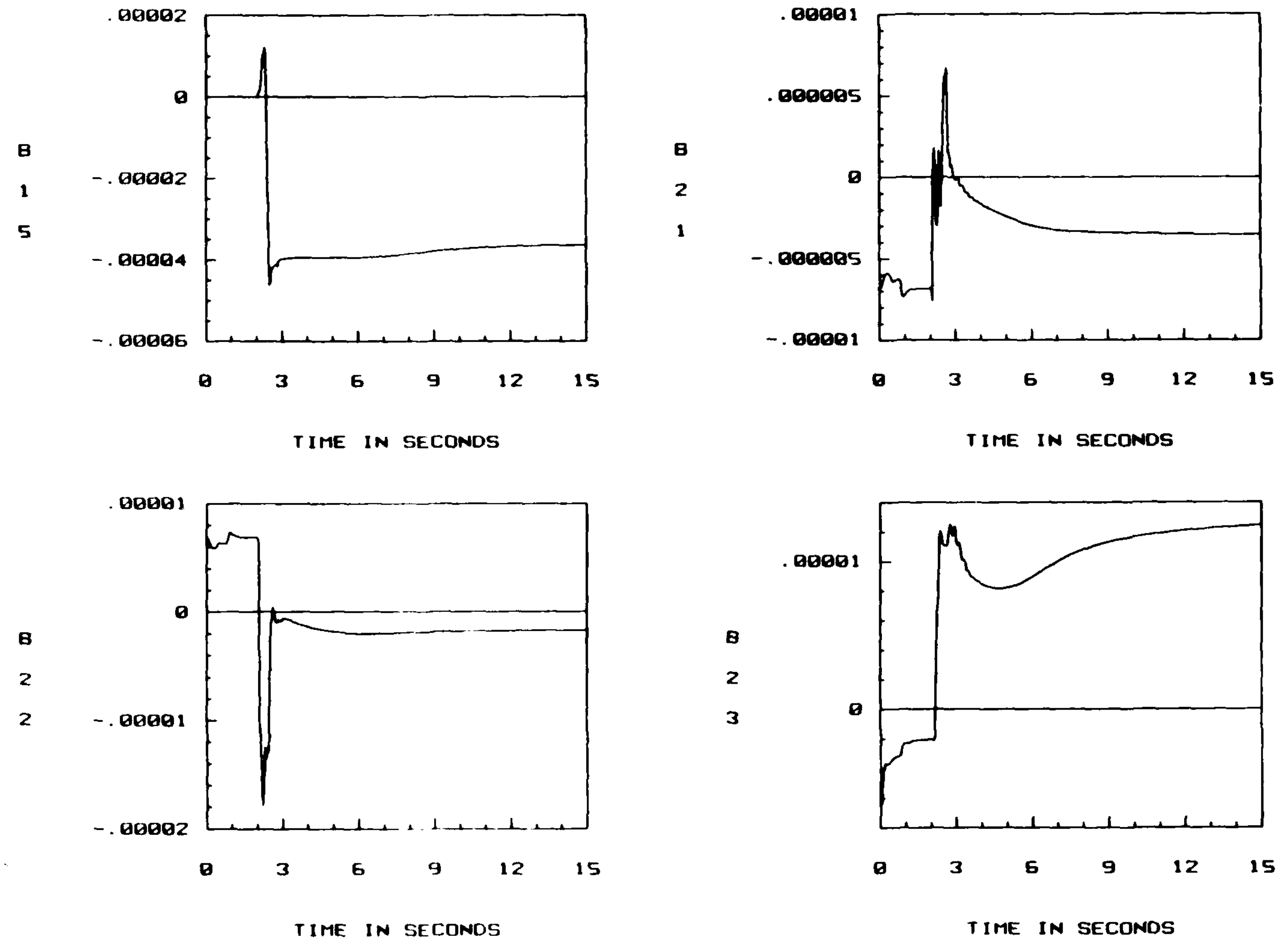


Figure C.16. B_1 ARMA Coefficients - $B_{15}, B_{21}, B_{22}, B_{23}$

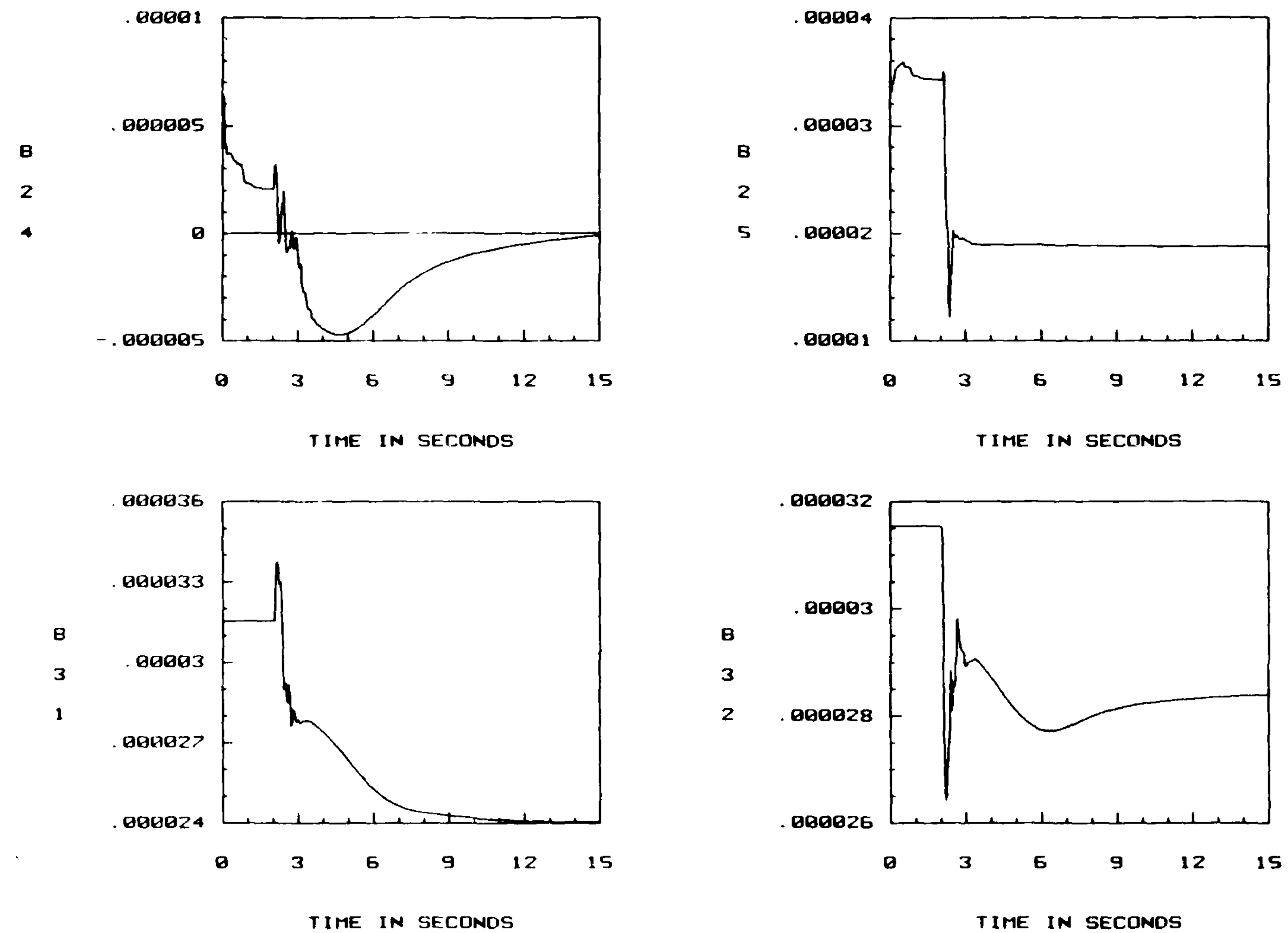


Figure C.17. B_1 ARMA Coefficients - $B_{24}, B_{25}, B_{31}, B_{32}$

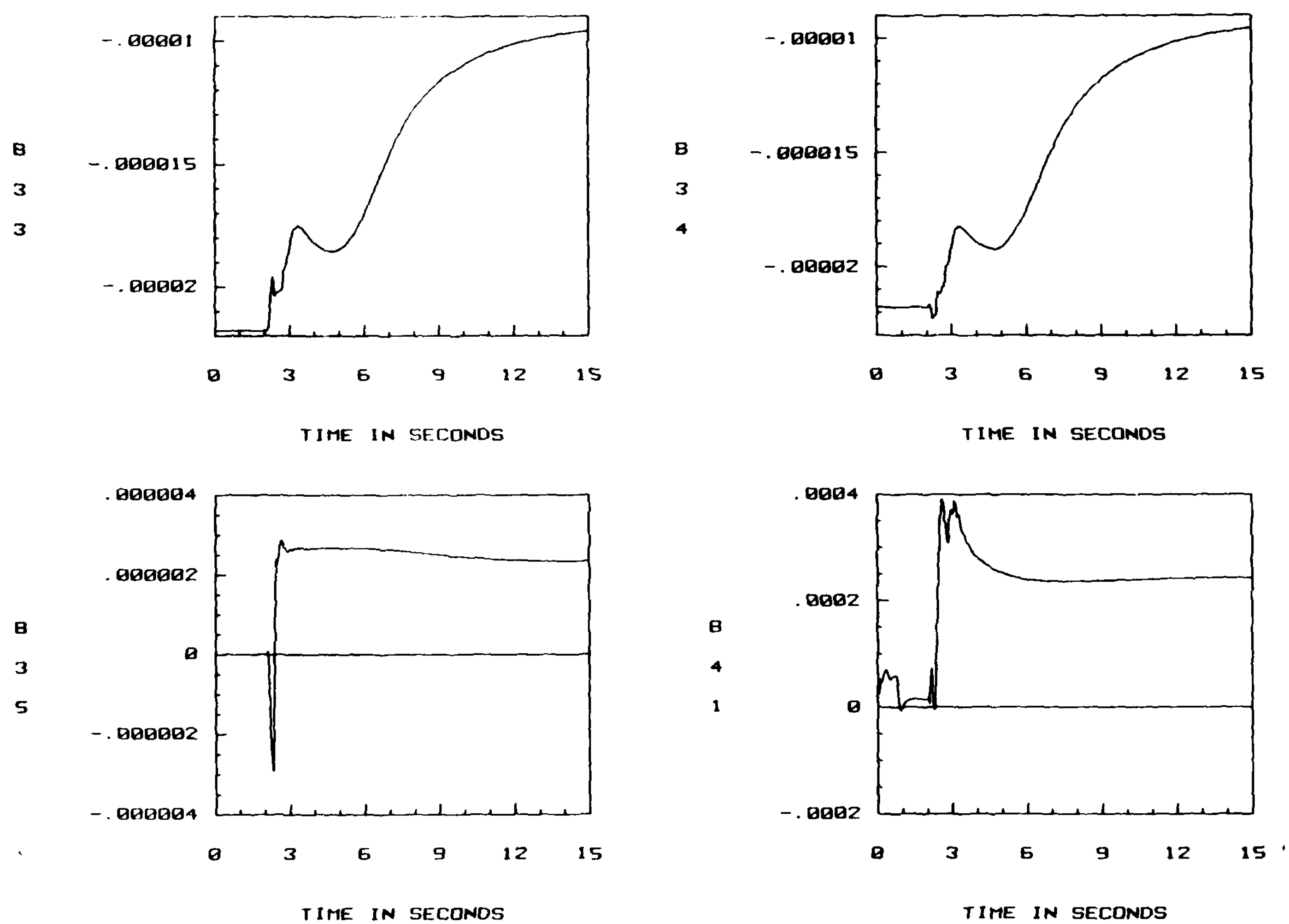


Figure C.18. B_1 ARMA Coefficients - $B_{33}, B_{34}, B_{35}, B_{41}$

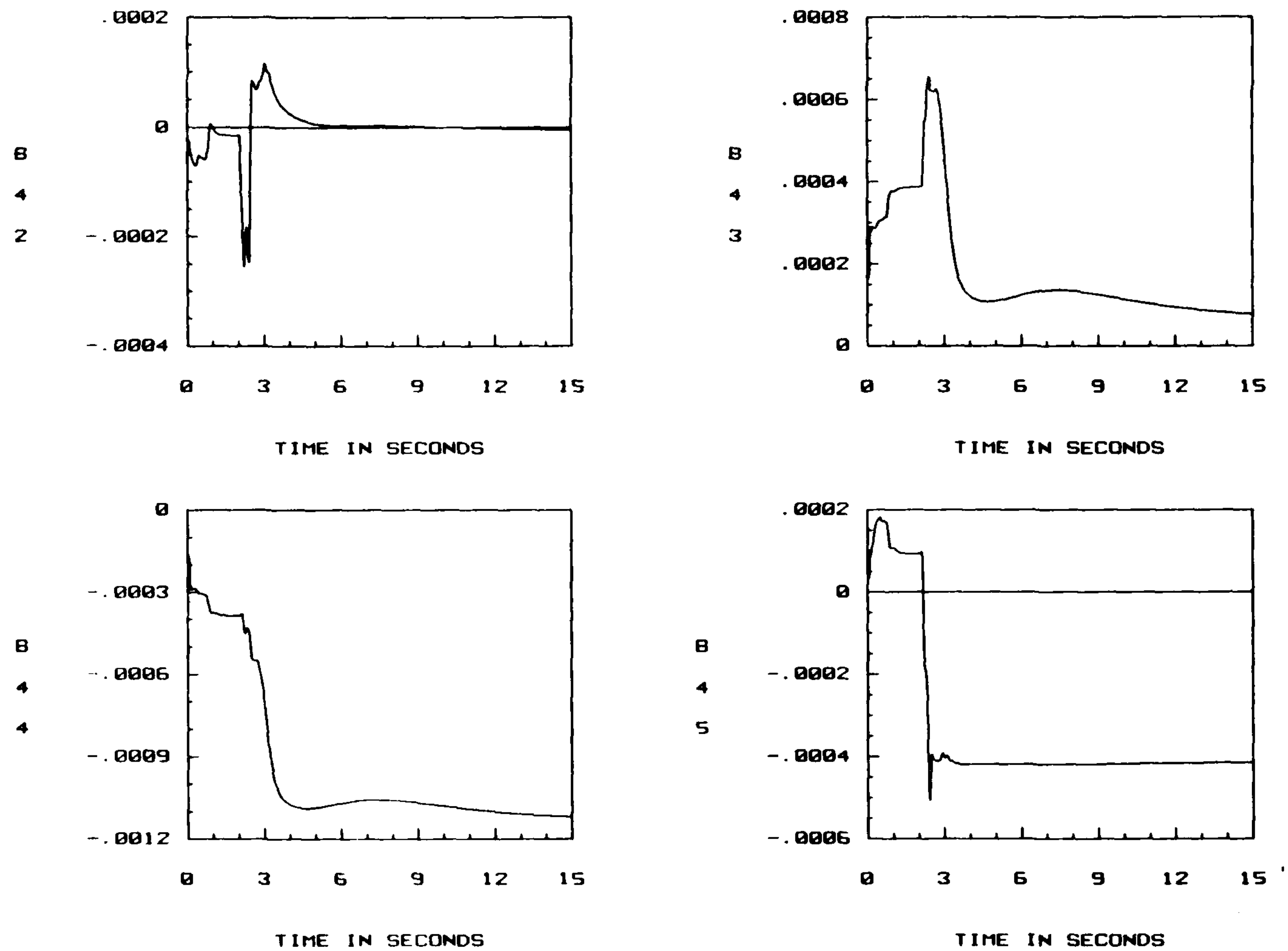


Figure C.19. B_1 ARMA Coefficients - $B_{42}, B_{43}, B_{44}, B_{45}$

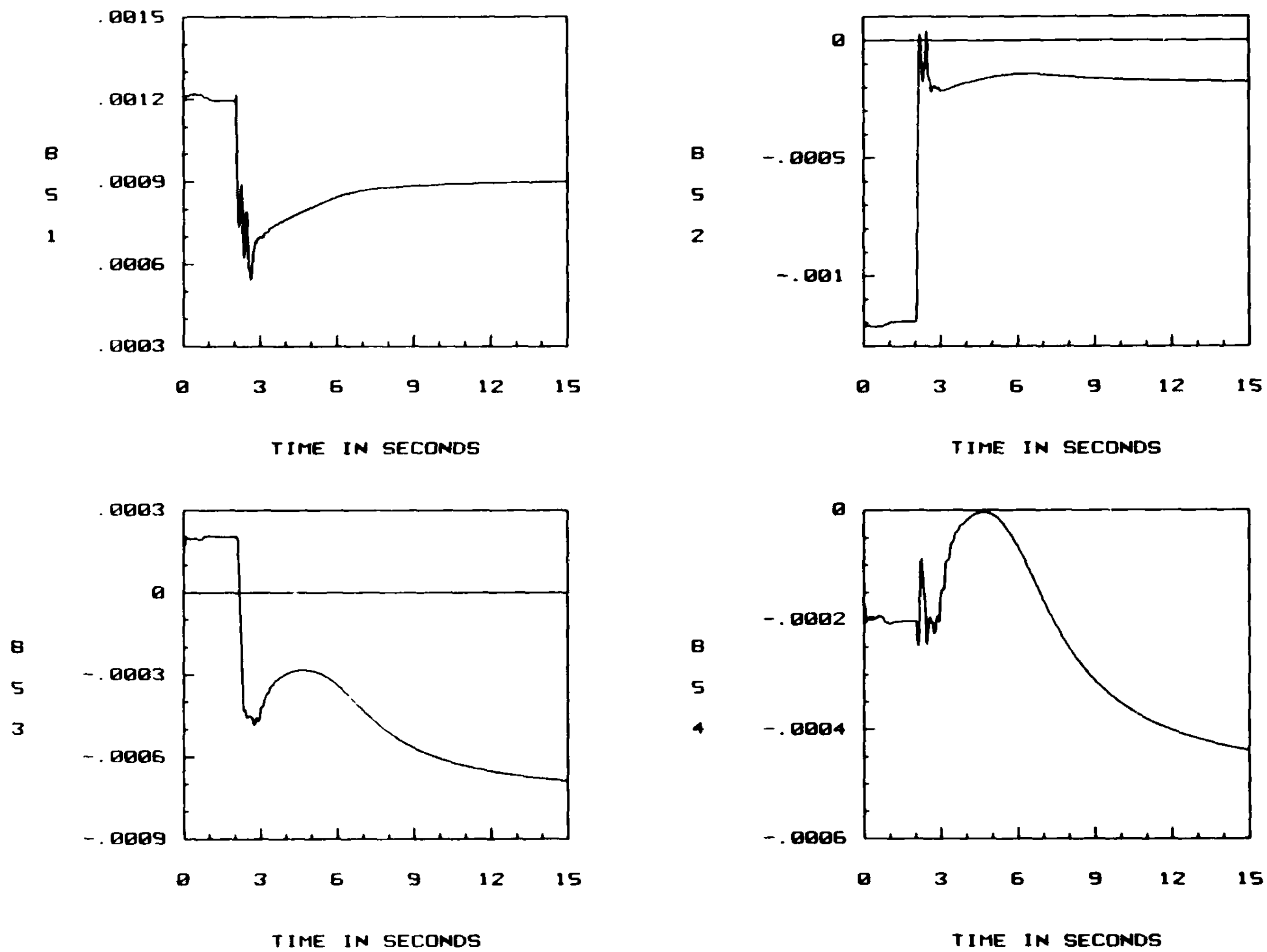


Figure C.20. B_1 ARMA Coefficients - $B_{51}, B_{52}, B_{53}, B_{54}$

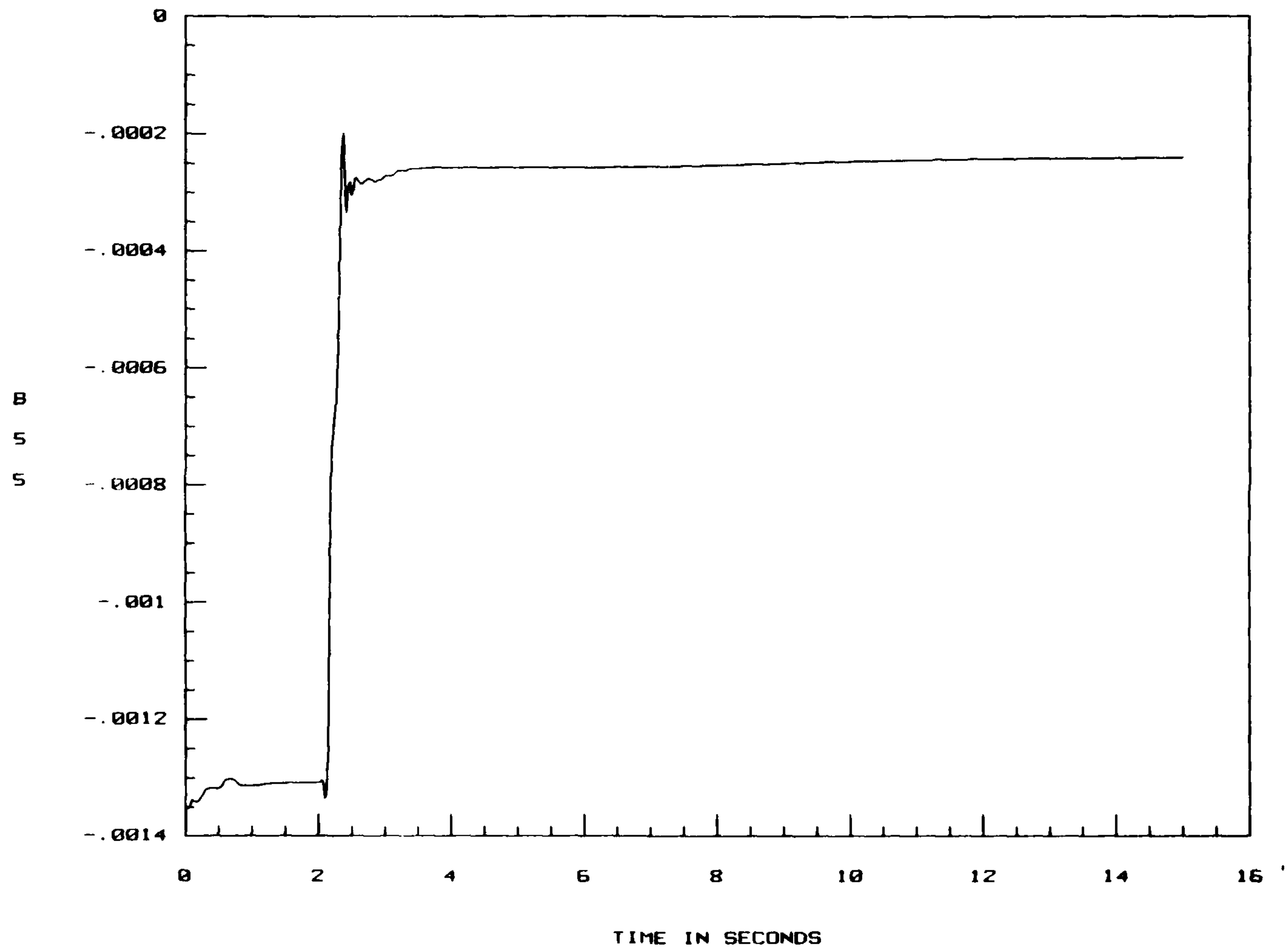


Figure C.21. B_1 ARMA Coefficients - B_{55}

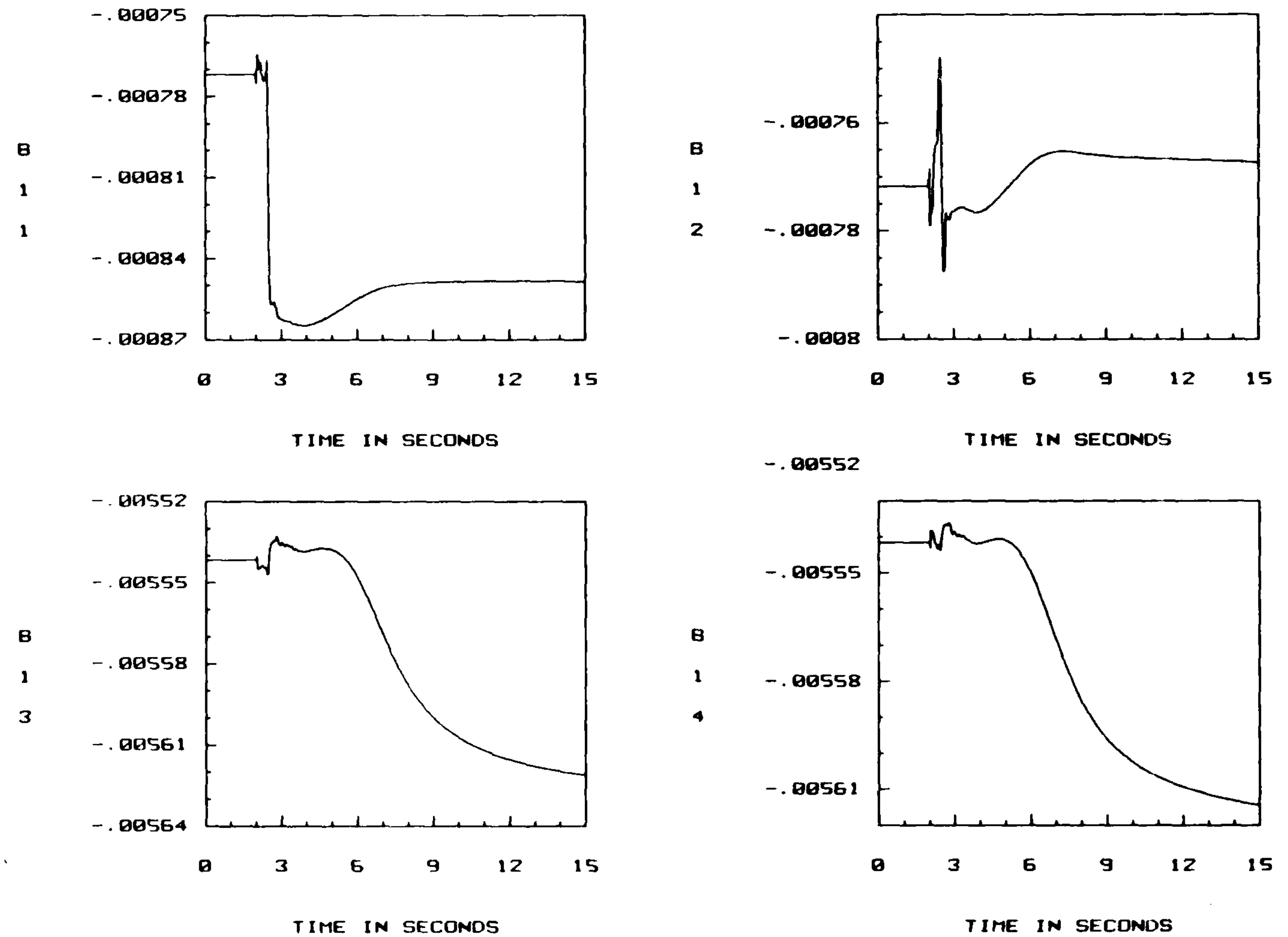


Figure C.22. B_2 ARMA Coefficients - $B_{11}, B_{12}, B_{13}, B_{14}$

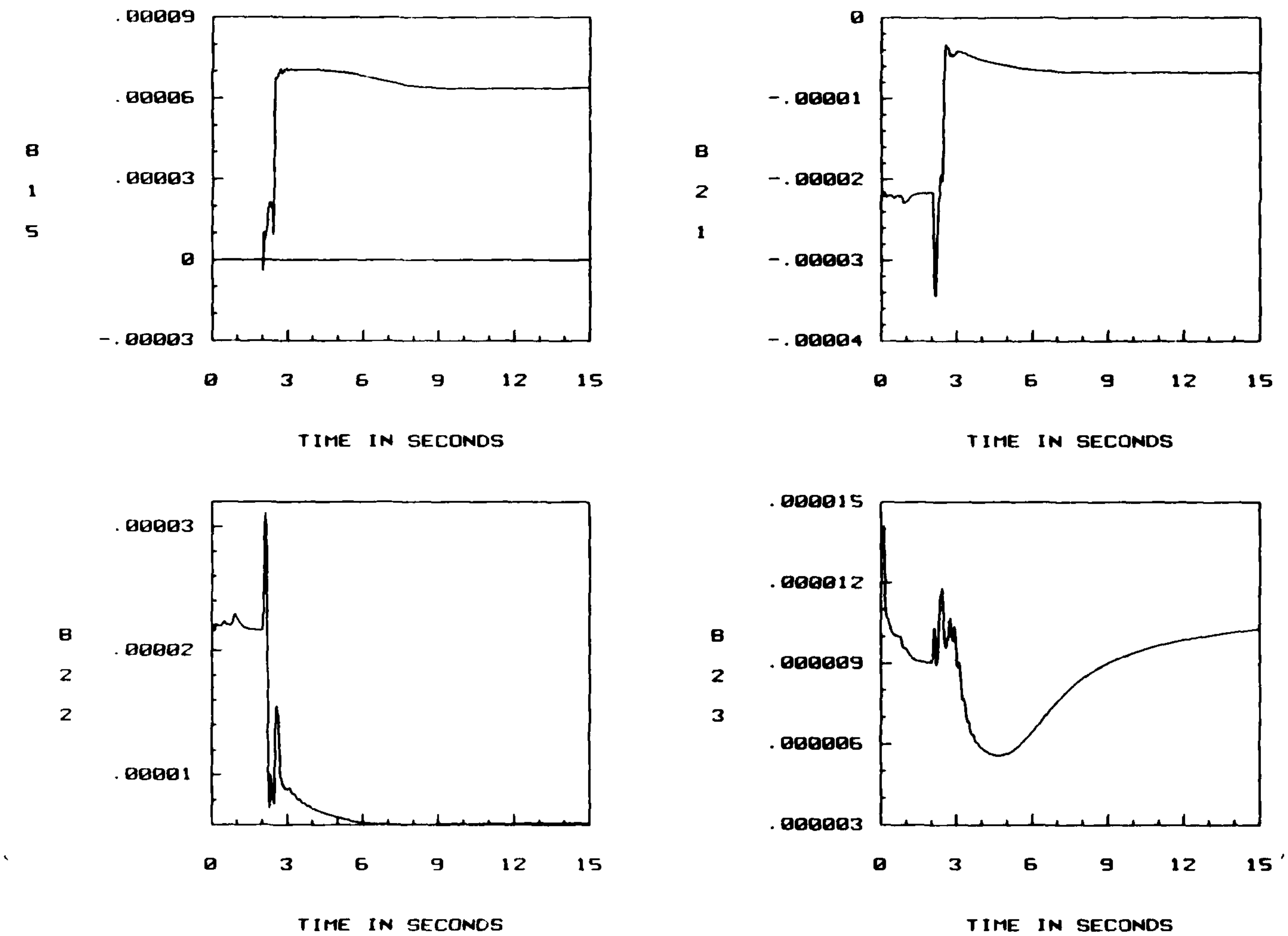


Figure C.23. B_2 ARMA Coefficients - $B_{15}, B_{21}, B_{22}, B_{23}$

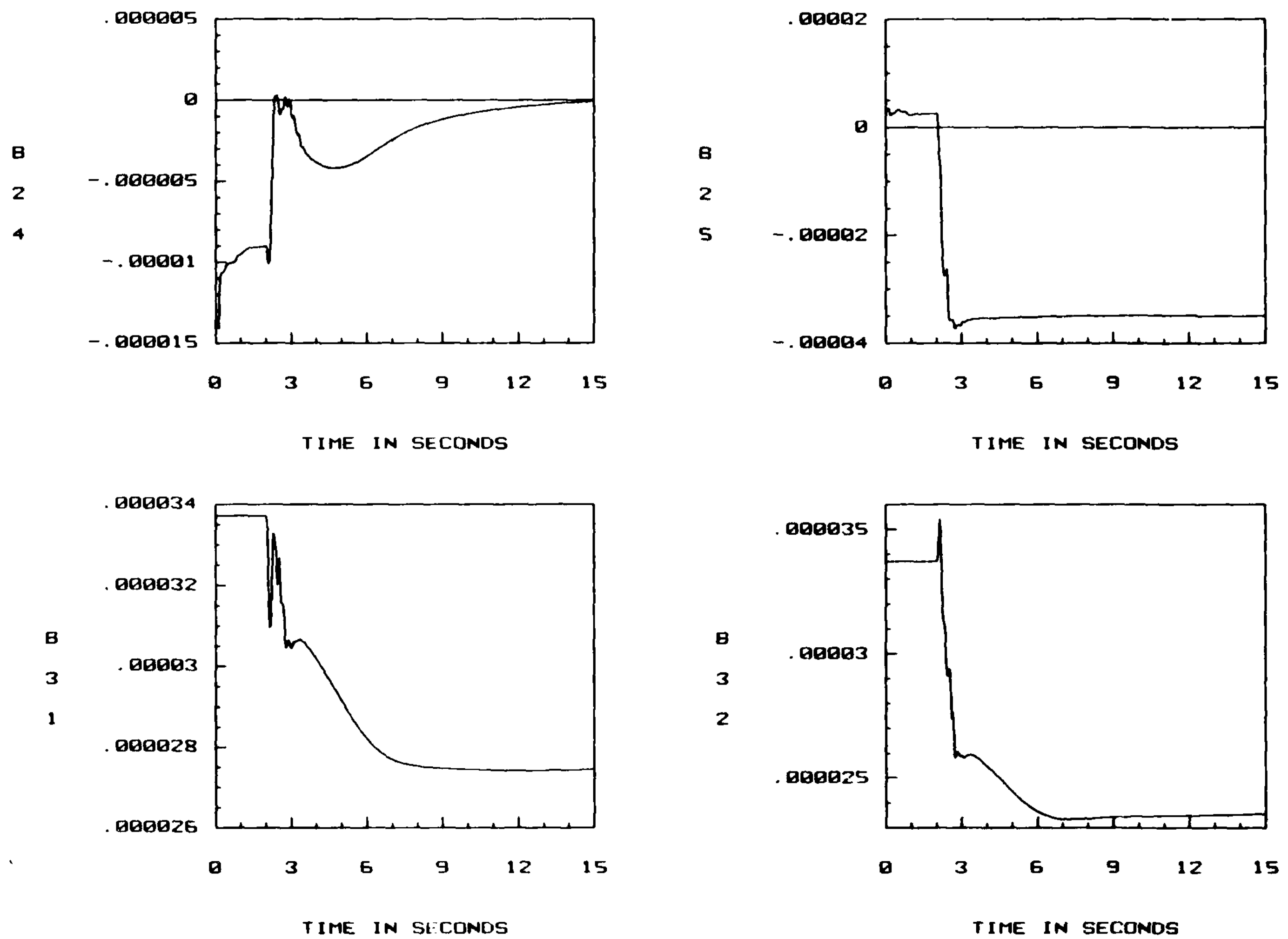


Figure C.24. B_2 ARMA Coefficients - $B_{24}, B_{25}, B_{31}, B_{32}$

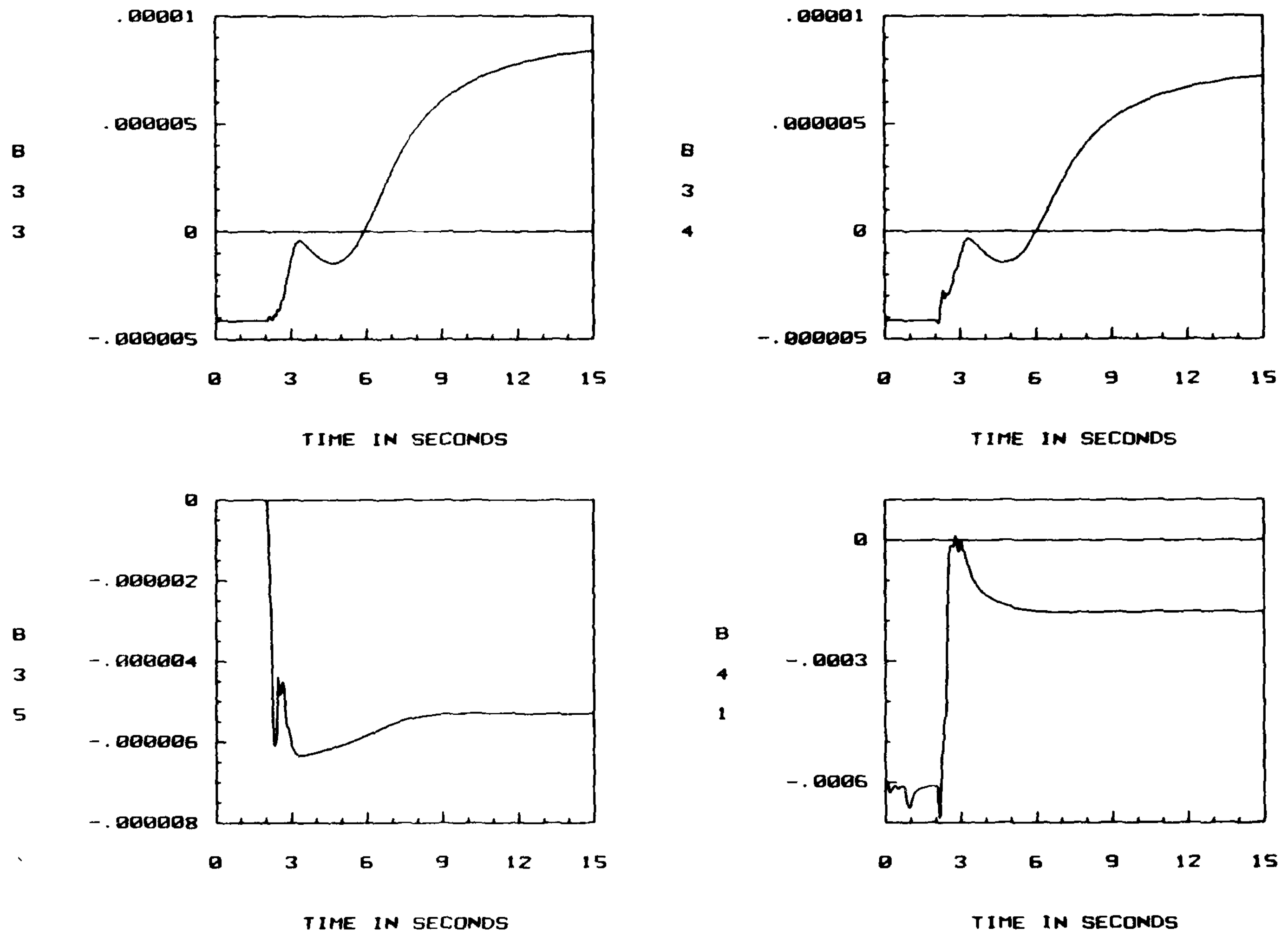


Figure C.25. B_2 ARMA Coefficients - $B_{33}, B_{34}, B_{35}, B_{41}$

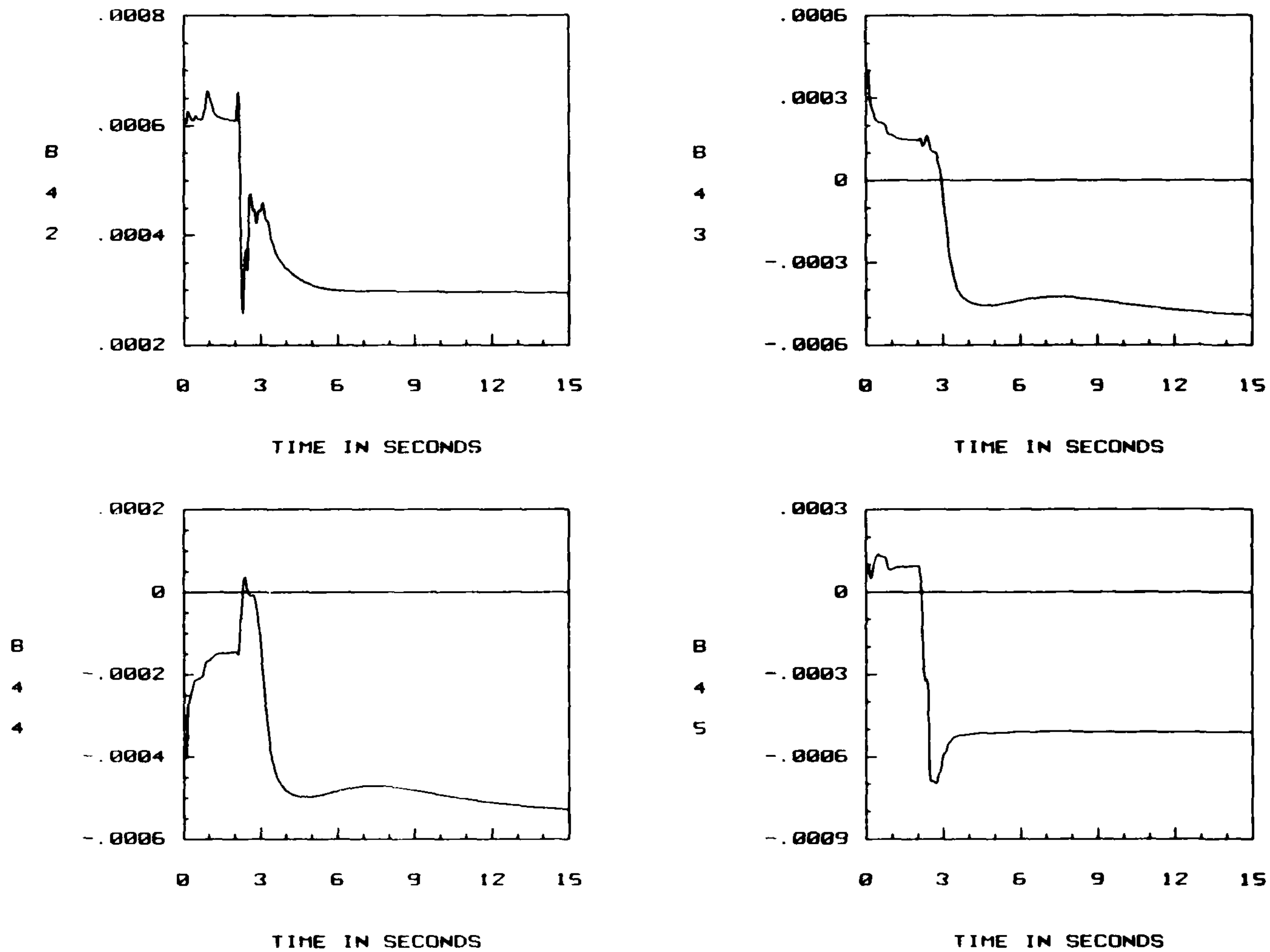


Figure C.26. B_2 ARMA Coefficients - $B_{42}, B_{43}, B_{44}, B_{45}$

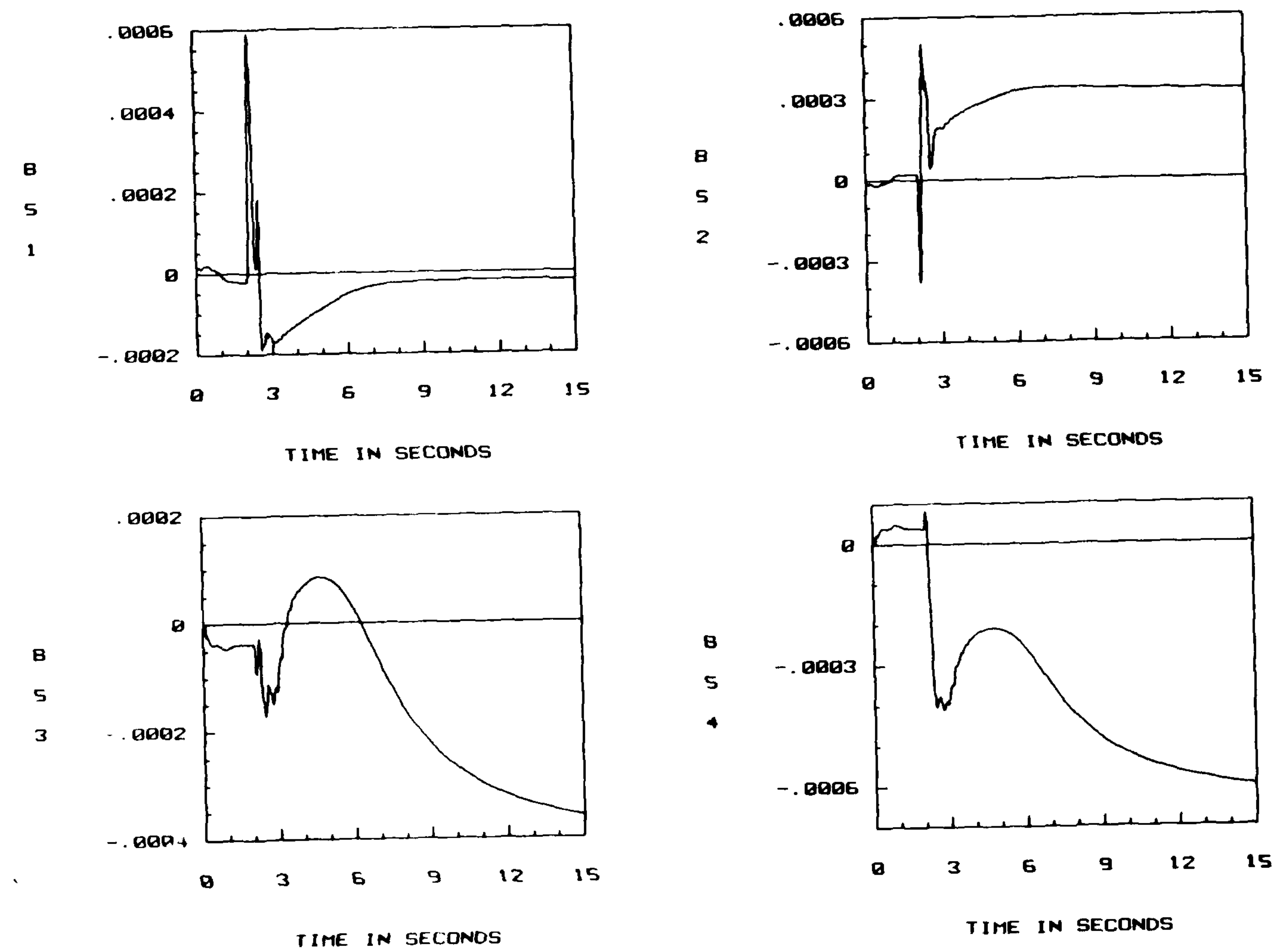


Figure C.27. B_2 ARMA Coefficients - $B_{51}, B_{52}, B_{53}, B_{54}$

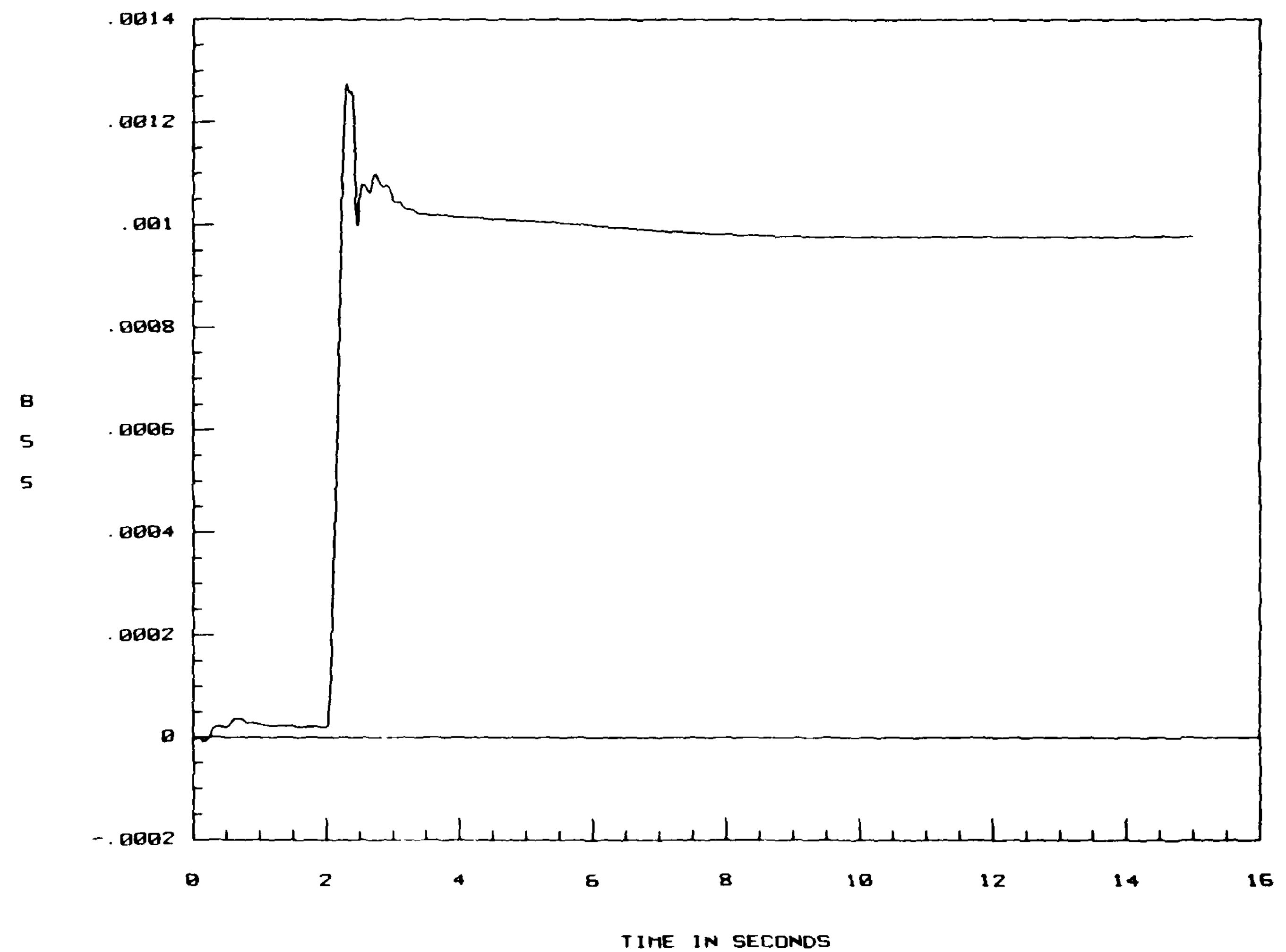


Figure C.28. B_2 ARMA Coefficients - B_{55}

Appendix D. *Porter's PID Control Law*

Introduction

This appendix contains a summary of the practical implementation of the PID control law as described by Professor Brian Porter of the University of Salford, England, and the application of the control law for the CRCA aircraft [20,21,24].

The PID design is satisfactory for the commanded 45 degree banked coordinated turn maneuver for the ACM Entry flight condition. Results for this condition are included.

PID Control Law

The PID control system of Figure D.1 is developed to provide a controller design for the irregular plant without the use of the measurement matrix, M. An equivalent PID control design, Figures D.2 and D.3, uses only output feedback when a plant is irregular.

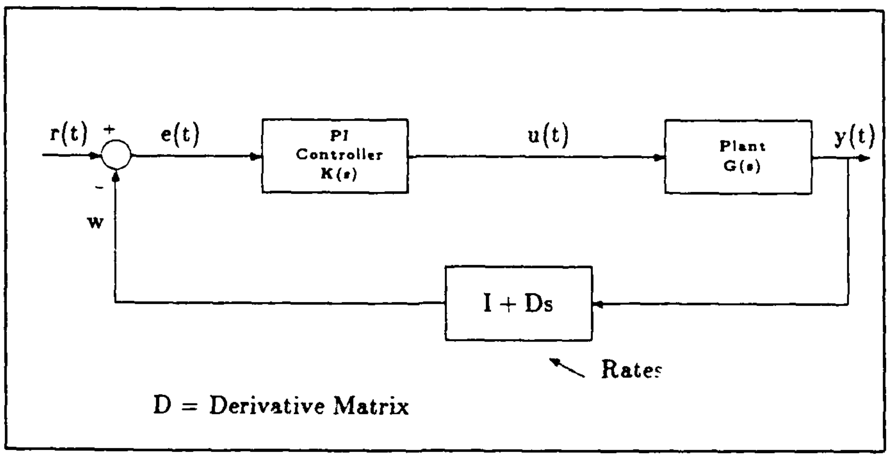


Figure D.1. PID Control
[24]

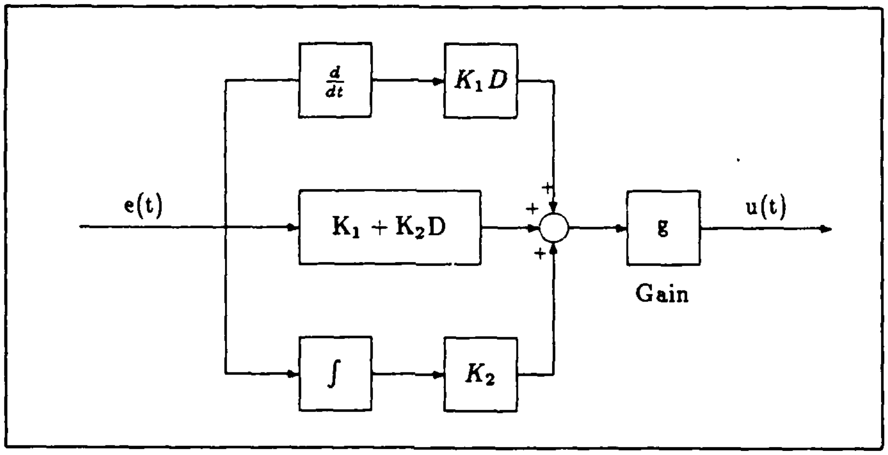


Figure D.2. Continuous PID Controller

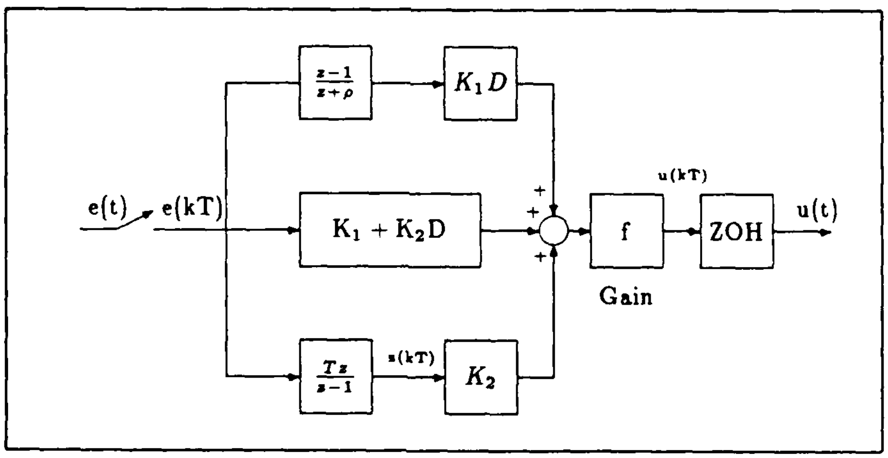


Figure D.3. Discrete PID Controller
[24]

Recall that the purpose of the measurement matrix M is to stabilize the system by rate feedback of the corresponding deficient first Markov parameter. The complete mathematical development of the PID controller is contained in references [20,21] and [24]. In his thesis, Sheldon showed that the introduction of a derivative matrix D, properly proportioned and located, would eliminate the need for the measurement matrix M [24, 52-58]. For the analysis to be valid, system equivalency between Figure 3.2 and Figure D.1 must be maintained. The relationships of Figure 3.2 and Equation 3.11 have the form

$$w(t) = \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

From Figure D.1,

$$\begin{aligned} w(t) &= y(t) + D\dot{y}(t) && \text{(D.1)} \\ &= Cx(t) + DC[Ax(t) + Bu(t)] && \text{(D.2)} \\ &= (C + DCA)x(t) + DCBu(t) && \text{(D.3)} \end{aligned}$$

Equation 3.11 and Equation D.3 place the constraints on D as

$$\begin{aligned} DCB &= 0 && \text{(D.4)} \\ C + DCA &= \begin{bmatrix} F_1 & F_2 \end{bmatrix} && \text{(D.5)} \end{aligned}$$

To satisfy the requirements of Equation D.4 and Equation D.5 the plant outputs of Equation 2.2 are ordered such that

$$CB = \begin{bmatrix} E_1 & E_2 \\ 0_{p \text{ x } m} & 0_{p \text{ x } p} \end{bmatrix} \qquad (D.6)$$

where,

$E_1 = (m - p)$ x $(m - p)$ matrix

$E_2 = (m - p)$ x p matrix

p = rank defect of the first Markov parameter of the open loop plant

In order to satisfy Equation D.4, D must be of the form

$$D = \begin{bmatrix} 0_{(m-p) \text{ x } (m-p)} & D_2 \\ 0_{p \text{ x } (m-p)} & D_4 \end{bmatrix} \tag{D.7}$$

where,

D_2 = $(m-p)$ x p matrix

D_4 = p x p matrix

p = rank defect of the first Markov parameter of the open loop plant

Practical implementation of the PID controller requires modification of the block diagram of Figure D.1. Sheldon shows the procedure for transforming the block diagram of Figure D.1 into the more useful design of Figure D.2, Figure D.3, and Figure D.4 [24, 52-56].

Selection of the derivative matrix D, K_1, and K_2, is developed by Professor Porter in reference [20] and [21] in terms of the open-loop step-response matrix from Equation 3.28

$$H(T) = \int_0^T C exp(At) B \text{ dt}$$

and the associated matrix

$$J(T) = H(2T)H^{-1}(T) \tag{D.8}$$

$$= \begin{bmatrix} J_1(T) & J_2(T) \\ J_3(T) & J_4(T) \end{bmatrix} \tag{D.9}$$

where,

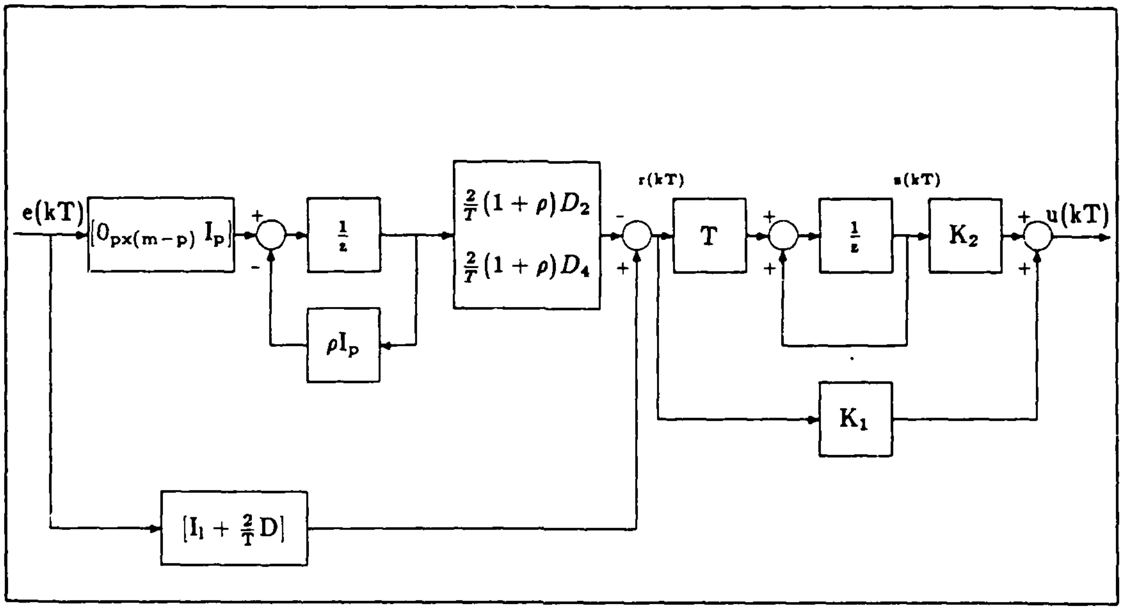


Figure D.4. PID Controller Implementation
[24]

$J_1(T) = (m-p) \text{ x } (m-p)$ matrix

$J_2(T) = (m-p) \text{ x } p$ matrix

$J_3(T) = p \text{ x } (m-p)$ matrix

$J_4(T) = p \text{ x } p$ matrix

Thus, the appropriate design parameters for the PID controller of Figure D.1 through Figure D.4 are

$$K_1 = TH(T)^{-1}\Sigma[TI_m + 2D]^{-1} \tag{D.10}$$

$$K_2 = \lambda K_1 \tag{D.11}$$

$$D_2 = \frac{\Sigma_1^{-1} J_2(T)\Sigma_2 D_4(T)}{(1+\alpha)} \tag{D.12}$$

$$D_4 = \delta I_p \tag{D.13}$$

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0_{(m-p) \times p} \\ 0_{p \times (m-p)} & \Sigma_2 \end{bmatrix} \tag{D.14}$$

$$= \begin{bmatrix} \sigma_1 I_{m-p} & 0_{(m-p) \times p} \\ 0_{p \times (m-p)} & \sigma_2 I_p \end{bmatrix} \tag{D.15}$$

where,

T = sampling time

H(T) = open loop step response matrix at time T

H(2T) = open loop step response matrix at 2 times sampling time

α = derivative filter constant

δ = derivative feedback gain

λ = ratio of integral to proportional control

Σ = diagonal weighting matrix = diag $[\sigma_1, \ldots, \sigma_m]$

The PID controller step response matrices H(T) and H(2T) are calculated from off-line tests of the open-loop stable plant or with an autoregressive algorithm for the open-loop unstable plant [5,20,21]. Using the development of the autoregressive parameter equivalency in Chapter 3, H(T) and H(2T) are represented by Equation 3.35

$$H(T) = B_1$$

$$H(2T) = B_1 + B_2 - A_1 B_1 \tag{D.16}$$

Design Procedure

The aircraft plant matrices of Table 2.4 and Table A.8 for the ACM Entry flight condition provide a basis for the design procedure. Recall that the rank deficiency of the first Markov parameter is two, $p = 2$. For convenience, Table A.8 is reproduced.

Table D.1. ACM Entry Plant Matrices

$$A = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0350 \\ -32.1804 & .0000 & -.0119 & -.0186 & -31.2350 & .0000 & .0000 & .0000 \\ -1.0634 & .0000 & -.0324 & -1.0634 & 894.4548 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0069 & -.6015 & .0000 & .0000 & .0000 \\ .0000 & .0360 & .0000 & .0000 & .0000 & -.0929 & .0349 & -.9994 \\ .0000 & .0000 & .0000 & .0000 & .0000 & -27.8066 & -2.0376 & .4913 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 2.4582 & -.0241 & -.4377 \end{bmatrix}$$

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0411 & .0411 & .1322 & .0866 & .1322 & .0866 & .1018 & .1018 & .0000 \\ -.3163 & -.3163 & -.9597 & -.6194 & -.9597 & -.6194 & -1.0183 & -1.0183 & .0000 \\ .1014 & .1014 & -.0284 & -.0215 & -.0284 & -.0215 & -.0200 & -.0200 & .0000 \\ .0003 & -.0003 & -.0002 & -.0001 & .0002 & .0001 & -.0001 & .0001 & .0006 \\ .0762 & -.0762 & .2219 & .2011 & -.2219 & -.2011 & .1109 & -.1109 & .1144 \\ .0486 & -.0486 & .0029 & .0021 & -.0029 & -.0021 & .0021 & -.0021 & -.0544 \end{bmatrix}$$

$$C = \begin{bmatrix} .0000 & .0000 & 1.0000 & .0349 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 & .0000 & .0000 \\ 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & 1.0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & .0000 & 1.0000 \end{bmatrix}$$

<u>B Matrix for Five Control Surfaces</u>

$$B = \begin{bmatrix} .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \\ .0411 & .0411 & .3206 & .3206 & .0000 \\ -.3163 & -.3163 & -2.5974 & -2.5974 & .0000 \\ .1014 & .1014 & -.0699 & -.0699 & .0000 \\ .0003 & -.0003 & -.0004 & .0004 & .0006 \\ .0762 & -.0762 & .5339 & -.5339 & .1144 \\ .0486 & -.0486 & .0071 & -.0071 & -.0544 \end{bmatrix}$$

$$x = \begin{bmatrix} \theta & \phi & u & w & q & \beta & p & r \end{bmatrix}^T \tag{D.17}$$

$$u = \begin{bmatrix} \delta_{cl} & \delta_{cr} & \delta_{ter} & \delta_{tel} & \delta_{rud} \end{bmatrix}^T \tag{D.18}$$

The matrix product CB is not in the proper form of Equation D.6, therefore, the C matrix rows are permutated to yield the new output vector for the PID design given by

$$\begin{aligned} v &= \text{forward velocity} \\ \beta &= \text{side slip angle} \\ r &= \text{yaw rate} \\ \theta &= \text{pitch angle} \\ \phi &= \text{bank angle} \end{aligned}$$

and the corresponding output equation is

$$\begin{bmatrix} v \\ \beta \\ r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0.0349 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ u \\ w \\ q \\ \beta \\ p \\ r \end{bmatrix} \tag{D.19}$$

The resulting matrix product is

$$CB = \left[\begin{array}{ccc|cc} .0300 & .0300 & .2299 & .2299 & .0000 \\ .0003 & -.0003 & -.0004 & .0004 & .0006 \\ .0486 & -.0486 & .0071 & -.0071 & -.0544 \\ \hline .0000 & .0000 & .0000 & .0000 & .0000 \\ .0000 & .0000 & .0000 & .0000 & .0000 \end{array}\right] \tag{D.20}$$

where,

$$E_1 = \begin{bmatrix} .0300 & .0300 & .2299 \\ .0003 & -.0003 & -.0004 \\ .0486 & -.0486 & .0071 \end{bmatrix} \tag{D.21}$$

$$E_2 = \begin{bmatrix} .2299 & .0000 \\ .0004 & .0006 \\ -.0071 & -.0544 \end{bmatrix} \qquad (D.22)$$

In order to satisfy the conditions of Equation D.4, the matrix product $DCB = 0$, the D matrix must be of the form given in Equation D.7 where,

$$D = \left[\begin{array}{ccc|cc} .0000 & .0000 & .0000 & D_{2_{11}} & D_{2_{12}} \\ .0000 & .0000 & .0000 & D_{2_{21}} & D_{2_{22}} \\ .0000 & .0000 & .0000 & D_{2_{31}} & D_{2_{32}} \\ \hline .0000 & .0000 & .0000 & D_{4_{11}} & D_{4_{12}} \\ .0000 & .0000 & .0000 & D_{4_{21}} & D_{4_{22}} \end{array}\right] \qquad (D.23)$$

The matrix D_4 is easily calculated from Equation D.13. For this design

$$\delta = 0.2 \qquad (D.24)$$

$$D_4 = \begin{bmatrix} 0.2 & 0.0 \\ 0.0 & 0.2 \end{bmatrix} \qquad (D.25)$$

The associated matrix J is calculated from Equation D.9 where H(T) and H(2T) is calculated from the ARMA model plant representation using Equations 3.35 and D.16. However, the rows of H(T) and H(2T) corresponding to the change in rows of the C matrix have to be interchanged to yield a new $\bar{H}(T)$ and $\bar{H}(2T)$, in double precision,

$$\bar{H}(T) = \begin{bmatrix} 7.3484D\text{-}4 & 7.3484D\text{-}4 & 5.8075D\text{-}3 & 5.8075D\text{-}3 & 0.0000D{+}0 \\ -6.7897D\text{-}6 & 6.7897D\text{-}6 & -6.3897D\text{-}6 & 6.3897D\text{-}6 & 3.3130D\text{-}5 \\ 1.2078D\text{-}3 & -1.2078D\text{-}3 & 1.7236D\text{-}4 & -1.7236D\text{-}4 & -1.3526D\text{-}3 \\ 3.1534D\text{-}5 & 3.1534D\text{-}5 & -2.1788D\text{-}5 & -2.1788D\text{-}5 & 0.0000D{+}0 \\ 2.4002D\text{-}5 & -2.4002D\text{-}5 & 1.6415D\text{-}4 & -1.6415D\text{-}4 & 3.4422D\text{-}5 \end{bmatrix} \qquad (D.26)$$

$$\bar{H}(2T) = \begin{bmatrix} 1.3518D\text{-}3 & 1.3518D\text{-}3 & 1.1789D\text{-}2 & 1.1789D\text{-}2 & 0.0000D{-}4 \\ -4.1875D\text{-}5 & 4.1875D\text{-}5 & -5.5901D\text{-}6 & 5.5901D\text{-}6 & 1.0213D\text{-}4 \\ 2.4000D\text{-}3 & -2.4000D\text{-}3 & 3.3507D\text{-}4 & -3.3507D\text{-}4 & -2.6890D\text{-}3 \\ 1.2560D\text{-}4 & 1.2560D\text{-}4 & -8.6983D\text{-}5 & -8.6983D\text{-}5 & 0.0000D{+}0 \\ 9.4836D\text{-}5 & -9.4836D\text{-}5 & 6.4579D\text{-}4 & -6.4579D\text{-}4 & 1.3462D\text{-}4 \end{bmatrix} \quad (D.27)$$

and

$$J(T) = \begin{bmatrix} 2.0147D{+}0 & 0.0000D{+}0 & 0.0000D{+}0 & -4.0800D{+}0 & 0.0000D{+}0 \\ 0.0000D{+}0 & 1.9965D{+}0 & -2.4834D\text{-}2 & 0.0000D{+}0 & 6.9735D\text{-}2 \\ 0.0000D{+}0 & 6.1296D\text{-}2 & 1.9884D{+}0 & 0.0000D{+}0 & -4.4197D\text{-}2 \\ -3.1351D\text{-}5 & 0.0000D{+}0 & 0.0000D{+}0 & 3.9839D{+}0 & 0.0000D{+}0 \\ 0.0000D{+}0 & -1.1297D\text{-}2 & 2.9373D\text{-}4 & 0.0000D{+}0 & 3.9333D{+}0 \end{bmatrix} \quad (D.28)$$

where,

$$J_1(T) = \begin{bmatrix} 2.0147D{+}0 & 0.0000D{+}0 & 0.0000D{+}0 \\ 0.0000D{+}0 & 1.9965D{+}0 & -2.4834D\text{-}2 \\ 0.0000D{+}0 & 6.1296D\text{-}2 & 1.9884D{+}0 \end{bmatrix} \quad (D.29)$$

$$J_2(T) = \begin{bmatrix} -4.0800D{+}0 & 0.0000D{+}0 \\ 0.0000D{+}0 & 6.9735D\text{-}2 \\ 0.0000D{+}0 & -4.4197D\text{-}2 \end{bmatrix} \quad (D.30)$$

$$J_3(T) = \begin{bmatrix} -3.1351D\text{-}5 & 0.0000D{+}0 & 0.0000D{+}0 \\ 0.0000D{+}0 & -1.1297D\text{-}2 & 2.9373D\text{-}4 \end{bmatrix} \quad (D.31)$$

$$J_4(T) = \begin{bmatrix} 3.9839D{+}0 & 0.0000D{+}0 \\ 0.0000D{+}0 & 3.9333D{+}0 \end{bmatrix} \quad (D.32)$$

The remaining design parameters,

$$\Sigma = \begin{bmatrix} 0.010 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.010 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.060 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.060 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.020 \end{bmatrix} \quad (D.33)$$

$$\lambda = 0.005 \tag{D.34}$$

$$\alpha = 0.4 \tag{D.35}$$

are selected from a trial and error process with the complete system simulation to yield a good output. Therefore, using Equations D.10 through D.12 and a sampling time of $T = 0.025$ sec,

$$K_1 = \begin{bmatrix} 5.4705D\text{-}1 & 2.1280D{+}0 & 3.1691D{+}1 & 6.0466D{+}1 & -1.9316D{+}1 \\ 5.4705D\text{-}1 & -2.1280D{+}0 & -3.1691D{+}1 & 6.0466D{+}1 & 1.9316D{+}1 \\ 7.9174D\text{-}1 & -6.9111D{+}1 & -5.7625D{+}0 & 6.5178D{+}0 & 9.8085D{+}0 \\ 7.9174D\text{-}1 & 6.9111D{+}1 & 5.7625D{+}0 & 6.5178D{+}0 & -9.8085D{+}0 \\ .00000D\text{-}0 & 3.6241D{+}1 & 1.0767D{+}1 & 0.0000D{+}0 & -3.2435D{+}1 \end{bmatrix} \tag{D.36}$$

$$K_2 = \begin{bmatrix} 2.7352D\text{-}3 & 1.0640D\text{-}0 & 1.5846D\text{-}1 & 3.0233D\text{-}1 & -9.6582D\text{-}2 \\ 2.7352D\text{-}3 & -1.0460D\text{-}0 & -1.5846D\text{-}1 & 3.0233D\text{-}1 & 9.6582D\text{-}2 \\ 3.9587D\text{-}3 & -3.4556D\text{-}1 & -2.8813D\text{-}2 & 3.2589D\text{-}2 & 4.9042D\text{-}2 \\ 3.9587D\text{-}3 & 3.4556D\text{-}1 & 2.8813D\text{-}2 & 3.2589D\text{-}2 & -4.9042D\text{-}2 \\ 0.0000D{+}0 & 1.8120D\text{-}0 & 5.3835D\text{-}2 & 0.0000D{+}0 & -2.6218D\text{-}1 \end{bmatrix} \tag{D.37}$$

$$D_2 = \begin{bmatrix} -3.49710 & 0.000000 \\ 0.00000 & 0.019924 \\ 0.00000 & -0.002105 \end{bmatrix} \tag{D.38}$$

$$\Sigma_1 = \begin{bmatrix} 0.01 & 0.0 & 0.0 \\ 0.0 & 0.01 & 0.0 \\ 0.0 & 0.0 & 0.06 \end{bmatrix} \tag{D.39}$$

$$\Sigma_2 = \begin{bmatrix} 0.06 & 0.00 \\ 0.00 & 0.02 \end{bmatrix} \tag{D.40}$$

Simulation

The roll and yaw rate commands for the PID controlled system for a coordinated turn are shown in Figure D.5 and D.6. The input models for these desired commands are given in Chapter 5.

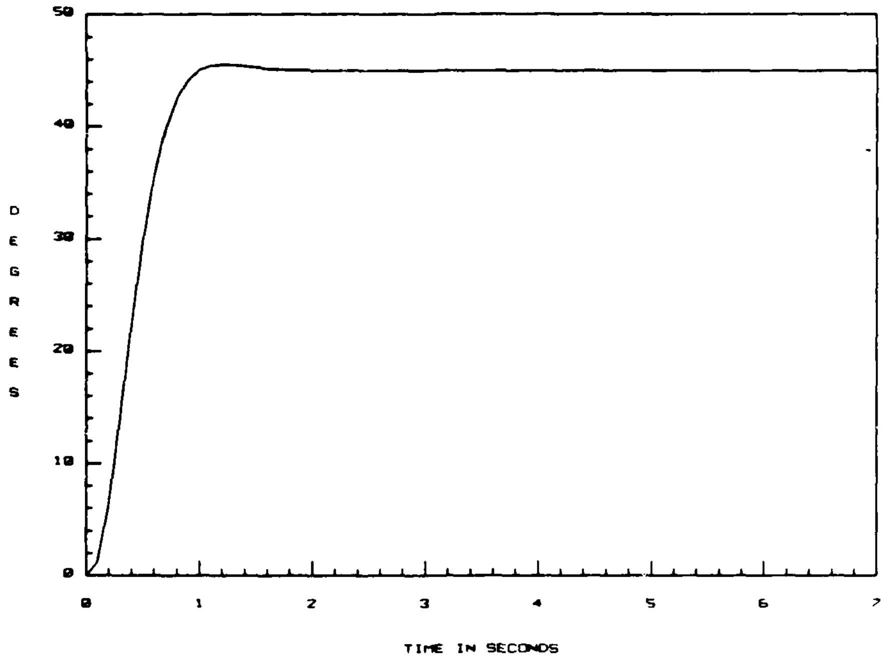


Figure D.5. ϕ_{cmd}

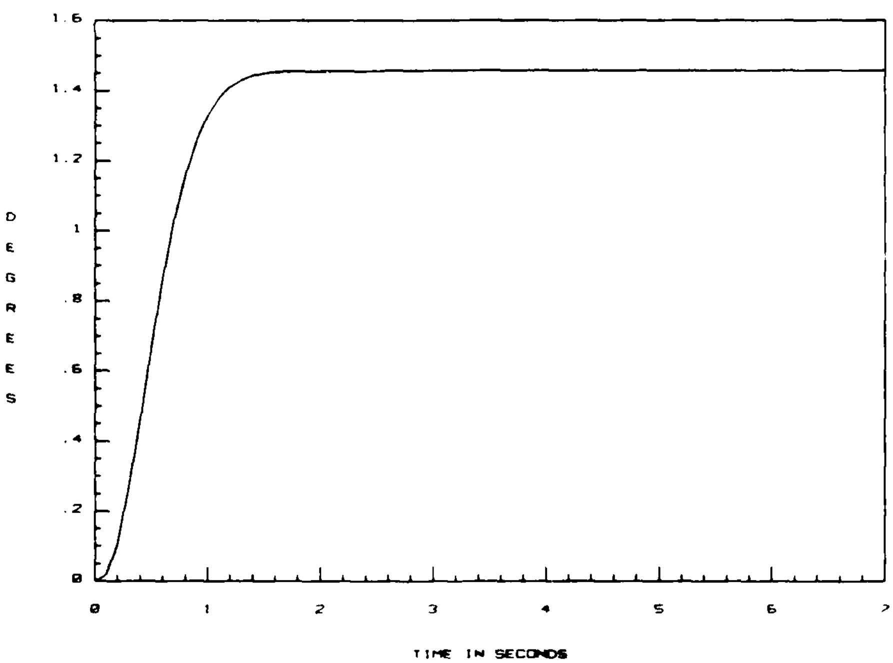


Figure D.6. r_{cmd}

The MATRIX$_x$ macros used in the simulation are listed in Appendix E. Chapter 4 develops the *relationship given* by Blakelock [4, 147] *for the roll angle* ϕ and the yaw rate r necessary for a coordinated turn. The yaw rate r is approximated by Equation 4.14

$$r\left(\frac{deg}{sec}\right) = \frac{g}{V}\sin(\phi)57.3\ \frac{deg}{rad}$$

where,

g = the gravitational constant $(32.174 ft/sec^2)$

V = the forward velocity of the aircraft

ϕ = the desired bank angle $(45° = .7854\ radians)$

For this flight condition,

$$r\left(\frac{deg}{sec}\right) = \frac{32.174\frac{ft}{sec^2}}{894.1\frac{ft}{sec}}\sin(45°)57.3\ \frac{deg}{rad} \tag{D.41}$$

$$\tag{D.42}$$

$$= 1.457\frac{deg}{sec} \tag{D.43}$$

Responses for the 45 degree coordinated turn maneuver are shown starting with Figure D.7; they illustrate the performance achieved with the PID controller.

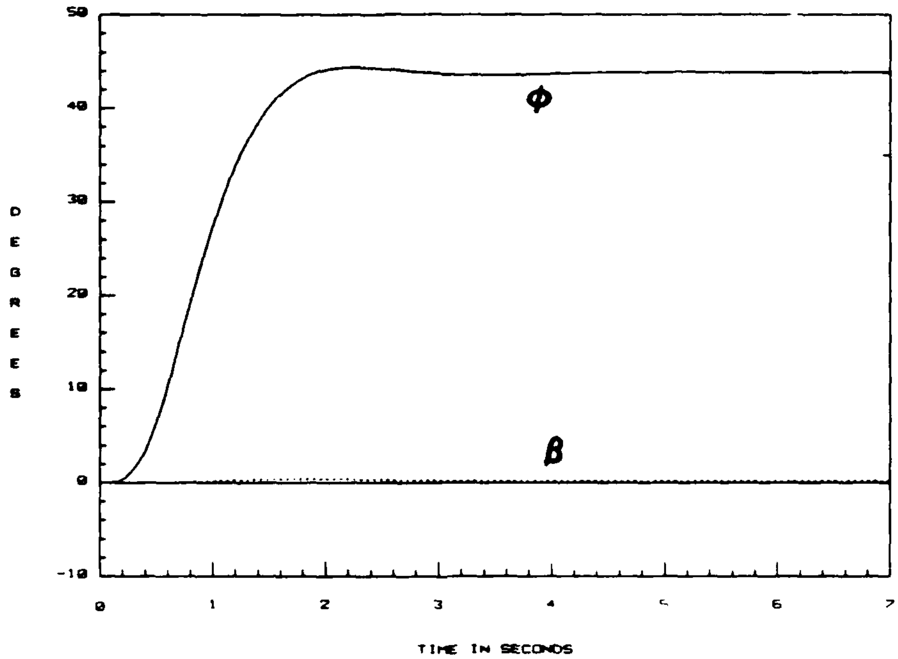


Figure D.7. ϕ and β - 45° Banked Turn

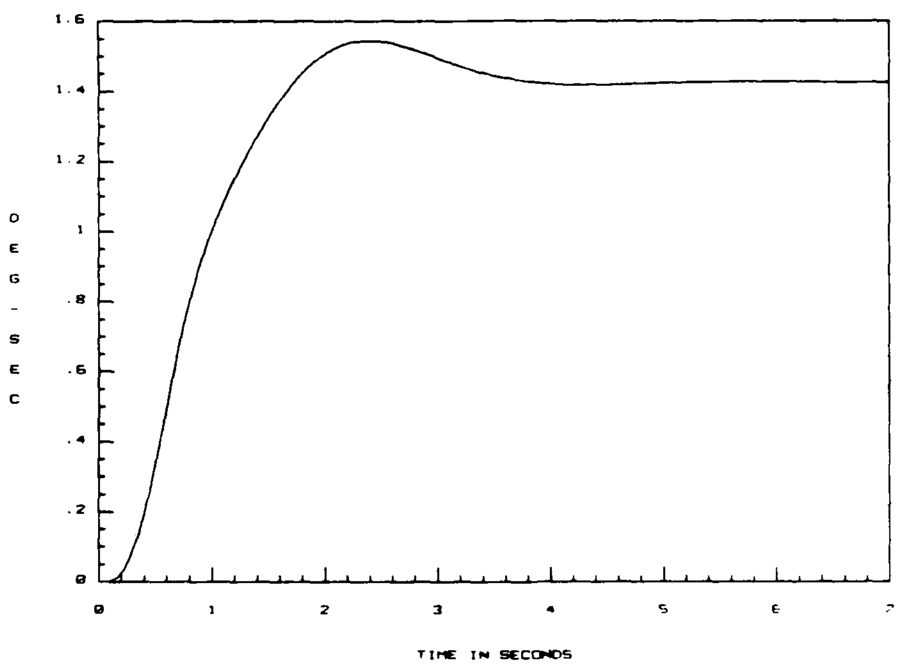


Figure D.8. r - 45° Banked Turn

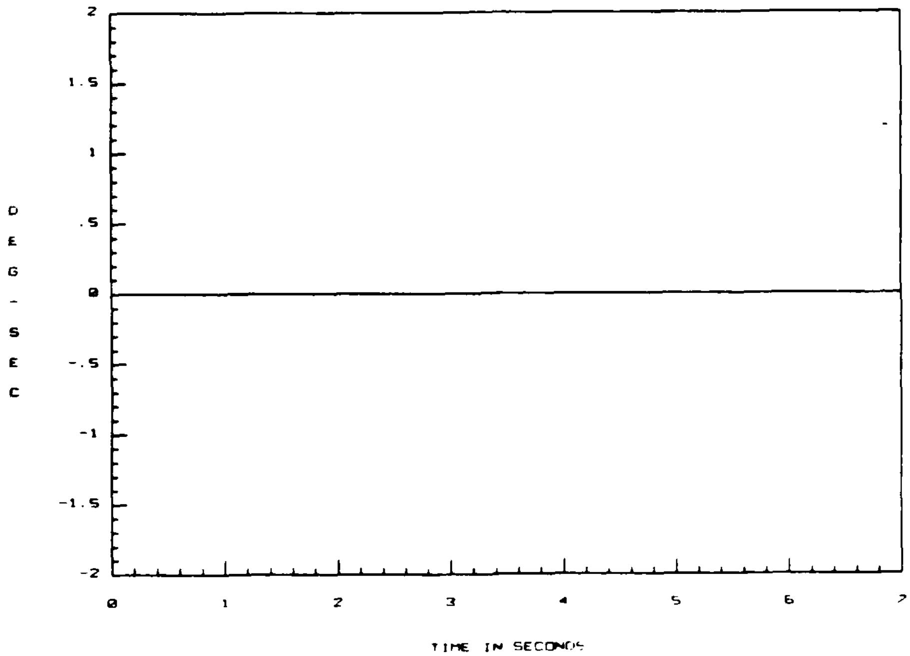


Figure D.9. u and w - 45° Banked Turn

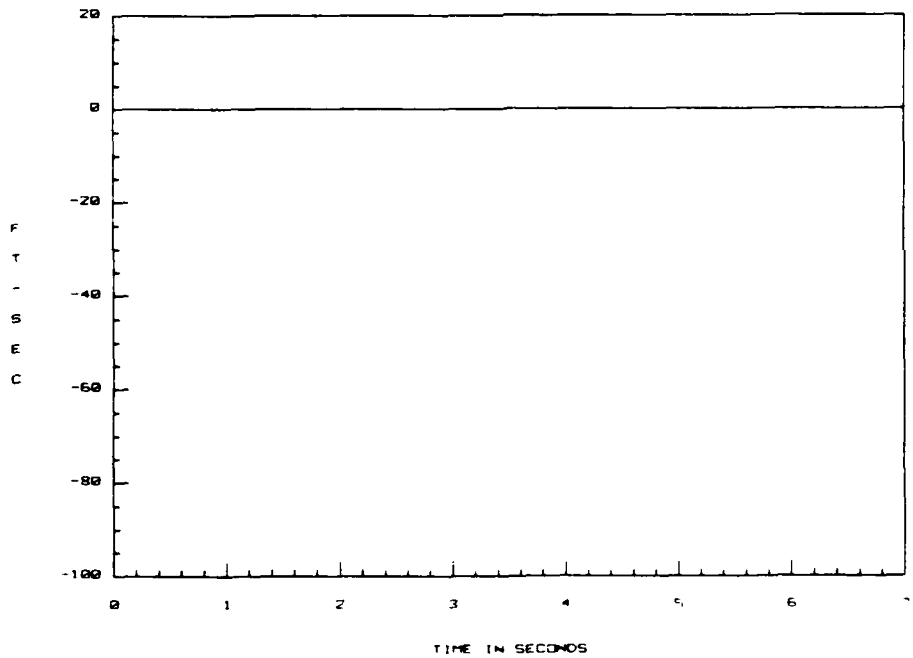


Figure D.10. θ and q - 45° Banked Turn

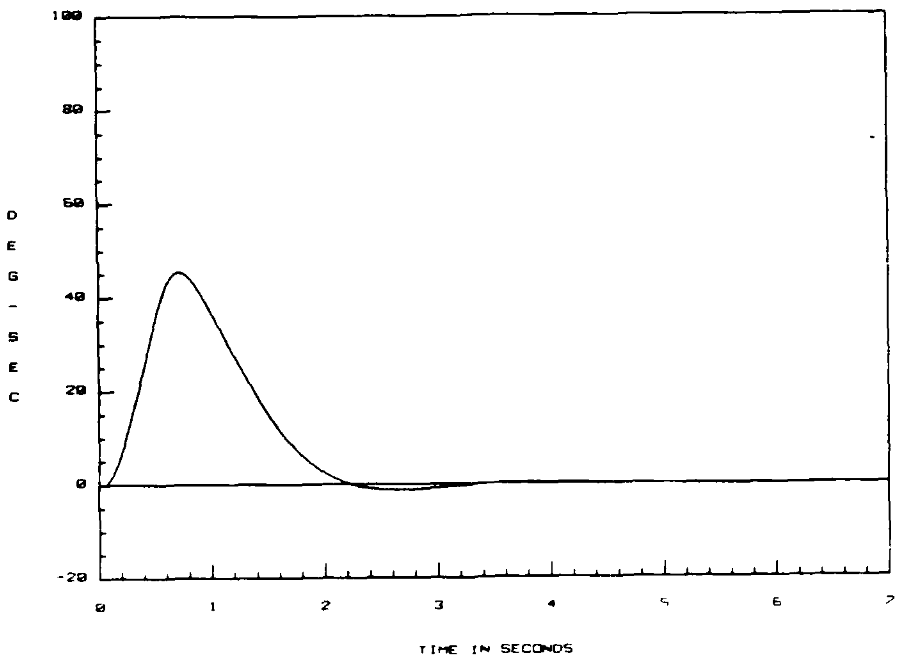


Figure D.11. p - 45° Banked Turn

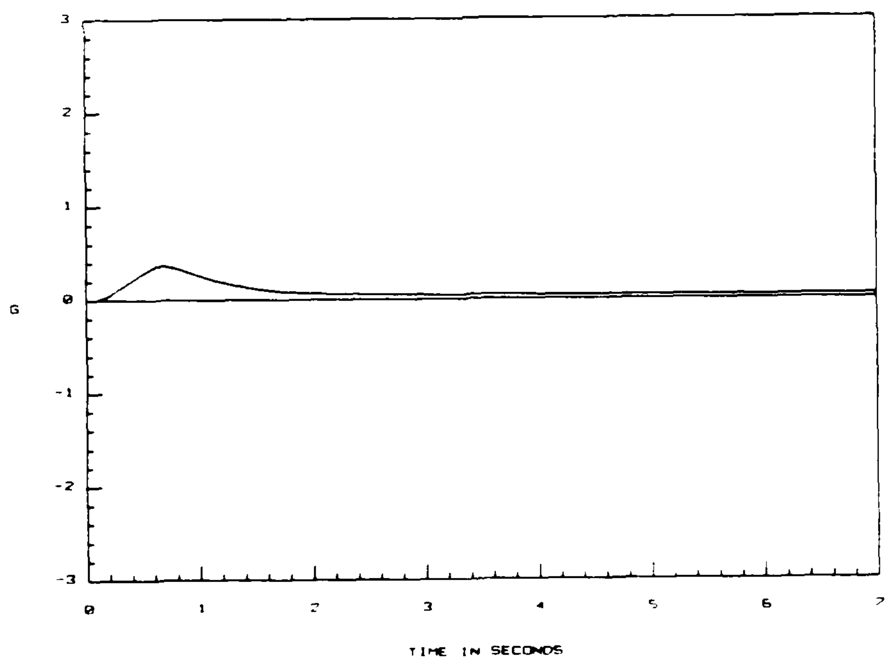


Figure D.12. Normal Acceleration - 45° Banked Turn

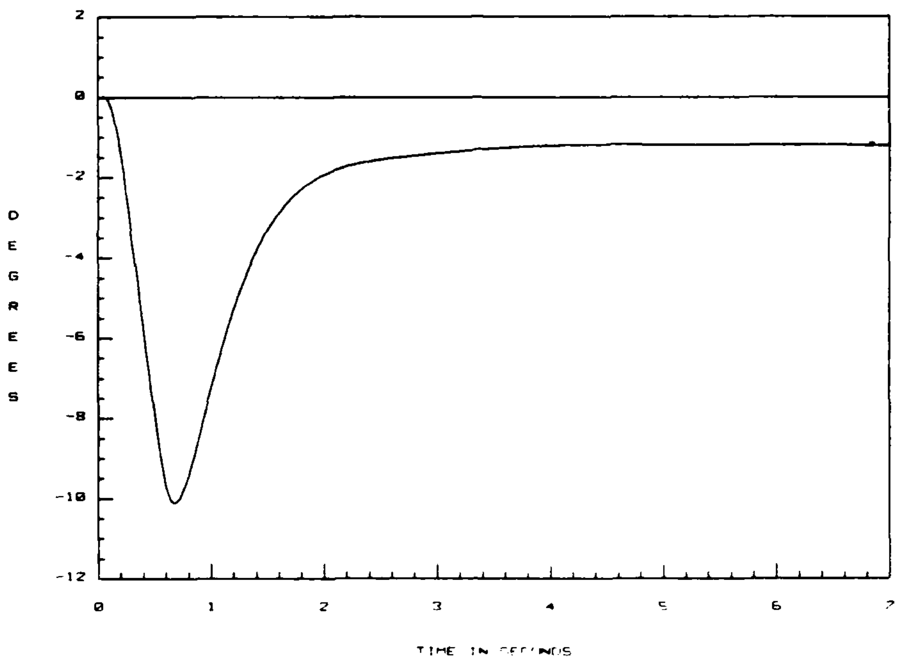


Figure D.13. Left Canard Deflection - 45° Banked Turn

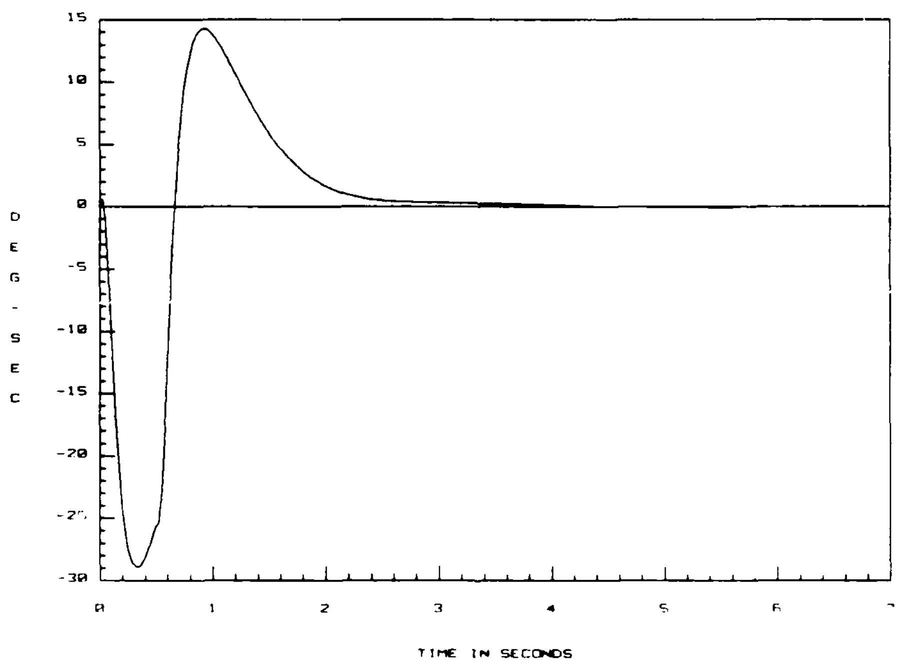


Figure D.14. Left Canard Deflection Rate - 45° Banked Turn

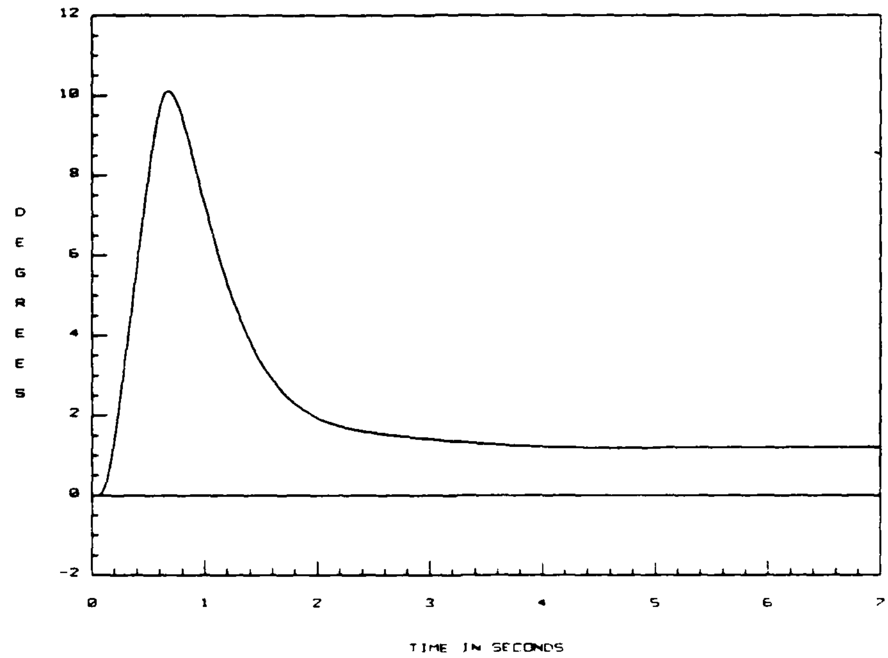


Figure D.15. Right Canard Deflection - 45° Banked Turn

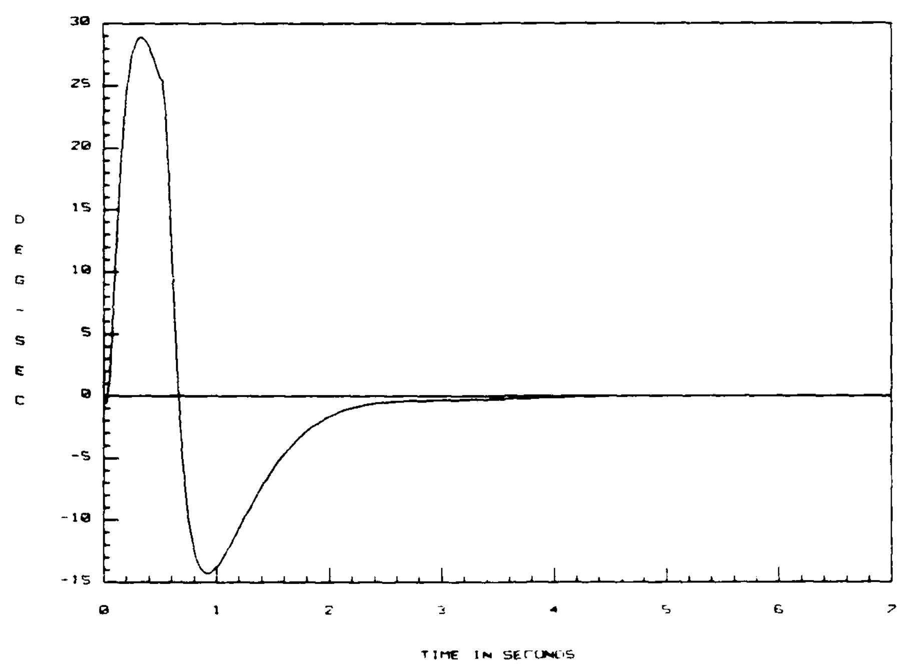


Figure D.16. Right Canard Deflection Rate - 45° Banked Turn

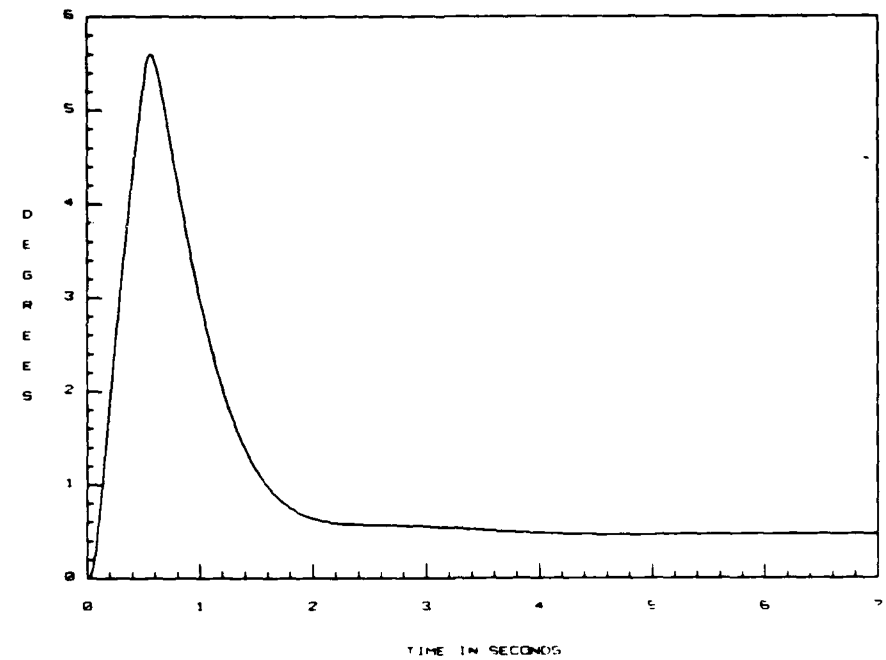


Figure D.17. Left Trailing Edge Deflection - 45° Banked Turn

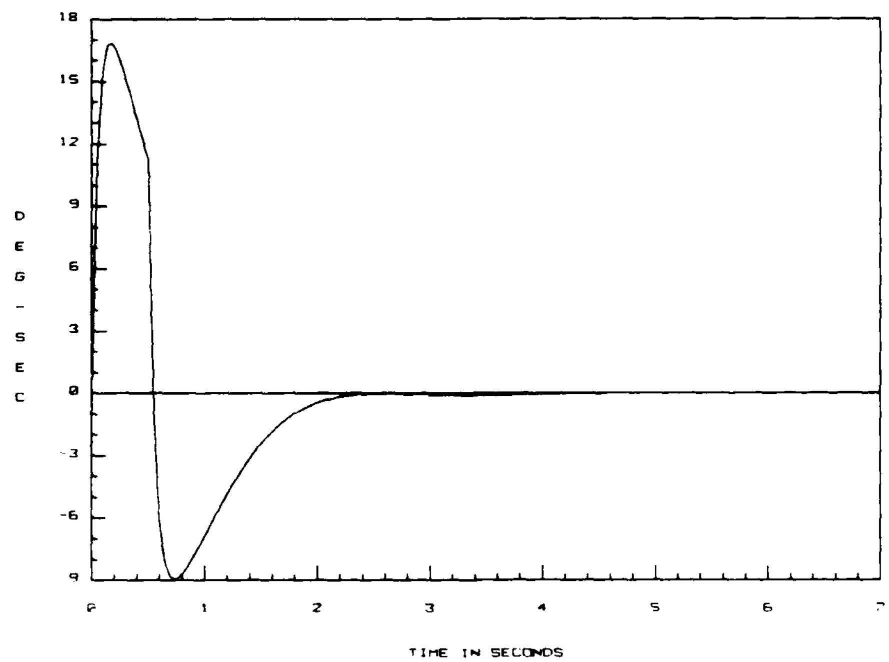


Figure D.18. Left Trailing Edge Deflection Rate - 45° Banked Turn

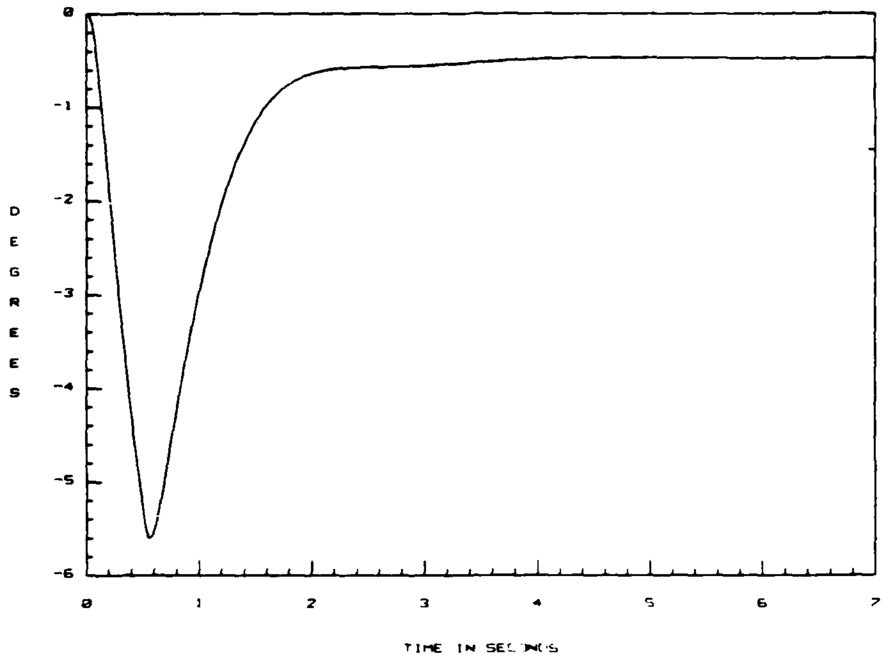


Figure D.19. Right Trailing Edge Deflection - 45° Banked Turn

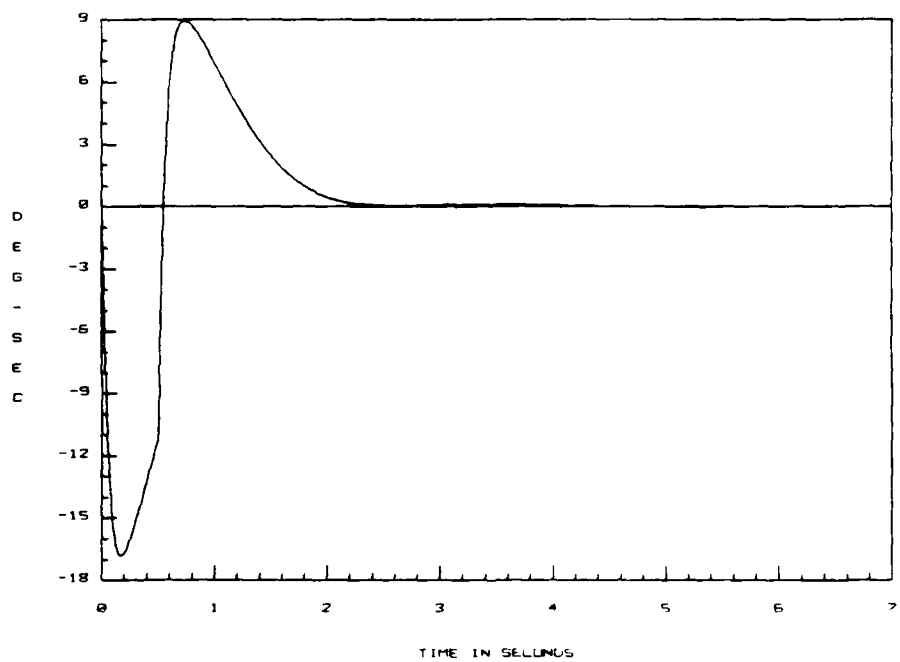


Figure D.20. Right Trailing Edge Deflection Rate - 45° Banked Turn

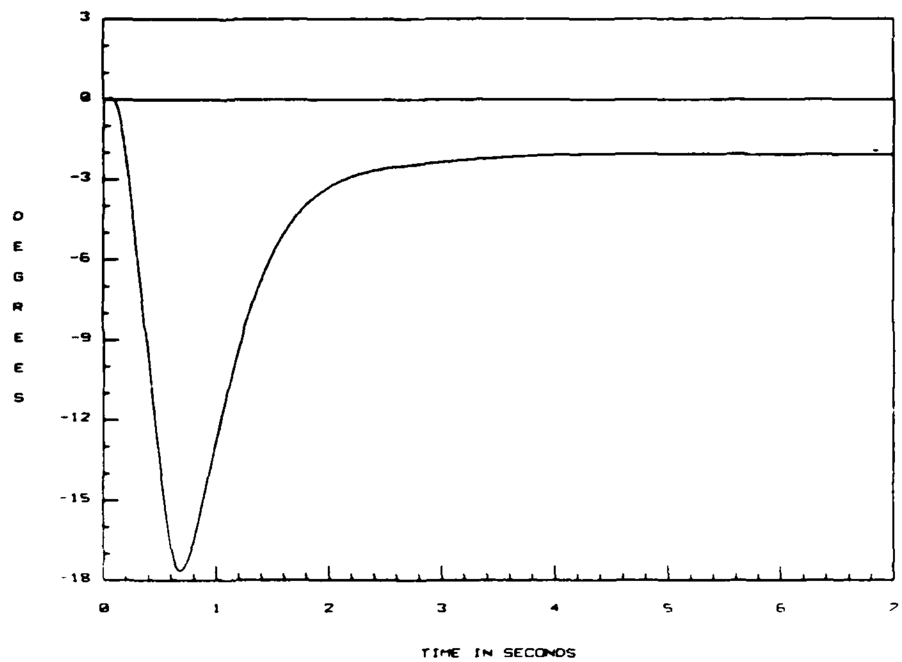


Figure D.21. Rudder Deflection - 45° Banked Turn

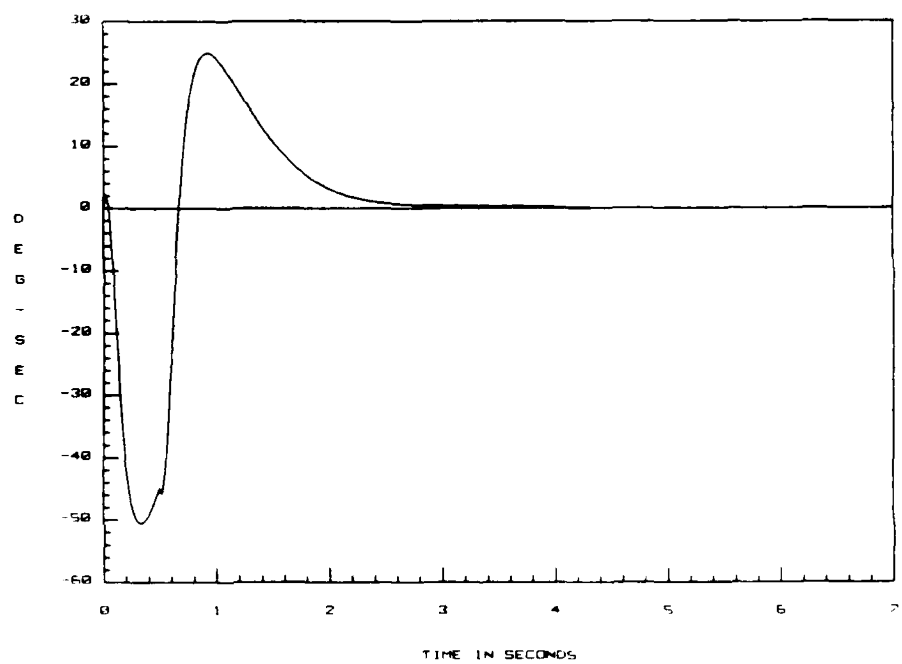


Figure D.22. Rudder Deflection Rate - 45° Banked Turn

Results and Analysis

The overall response of the CRCA with the PID controller is satisfactory, but is certainly not as robust as the PI controller. The overshoot on the bank angle is much larger and the response more sluggish for the ϕ and r commands. In addition, the overshoot is much harder to dampen and the corresponding elements of the Σ matrix are more sensitive to magnitude changes. Very little integral control is needed, as indicated by the value of $\lambda = 0.005$. Stabilizing the entire system is a tedious task with individual Σ element magnitudes having to be quite small. If the system had been compensated prior to the PID design to yield an equivalent open-loop stable plant, the PID controller design would have been more attractive. Since the plant dynamics are accessible, the use of the measurement matrix is a preferred approach and easily stabilizes the system during the PI control law implementation.

Appendix E. *ACM Entry State and Output Responses*

Introduction

This appendix provides a complete set of the $MATRIX_x$ simulation output responses for the nominal flight condition, ACM Entry. This includes the continuous and discrete PI controllers. The discrete controller output responses that are based upon gains calculated from the step-response matrices are indicated by the following entry in the figure caption: Discrete (Step-Response Matrix). Results for the remaining flight conditions and failure cases are available under separate cover. This appendix is divided into two parts. The first part shows the state and output responses obtained from ramped input commands while the second part provides a corresponding illustration of responses for the model following commands.

Ramped Input Commands

Responses for the 45 degree coordinated turn maneuver are shown starting with Figure E.1 and illustrate the performance of the controller.

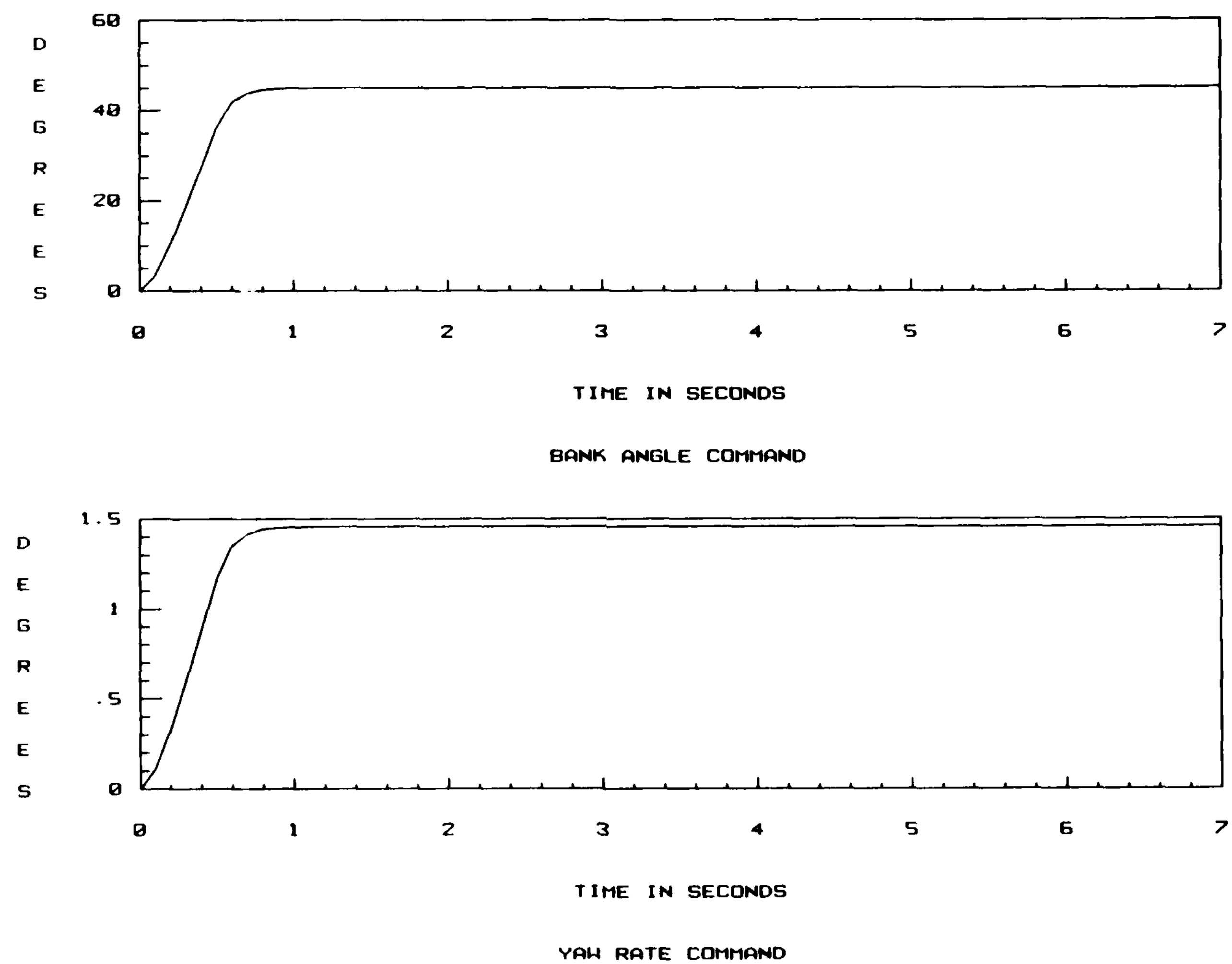


Figure E.1. ϕ_{cmd} and r_{cmd}

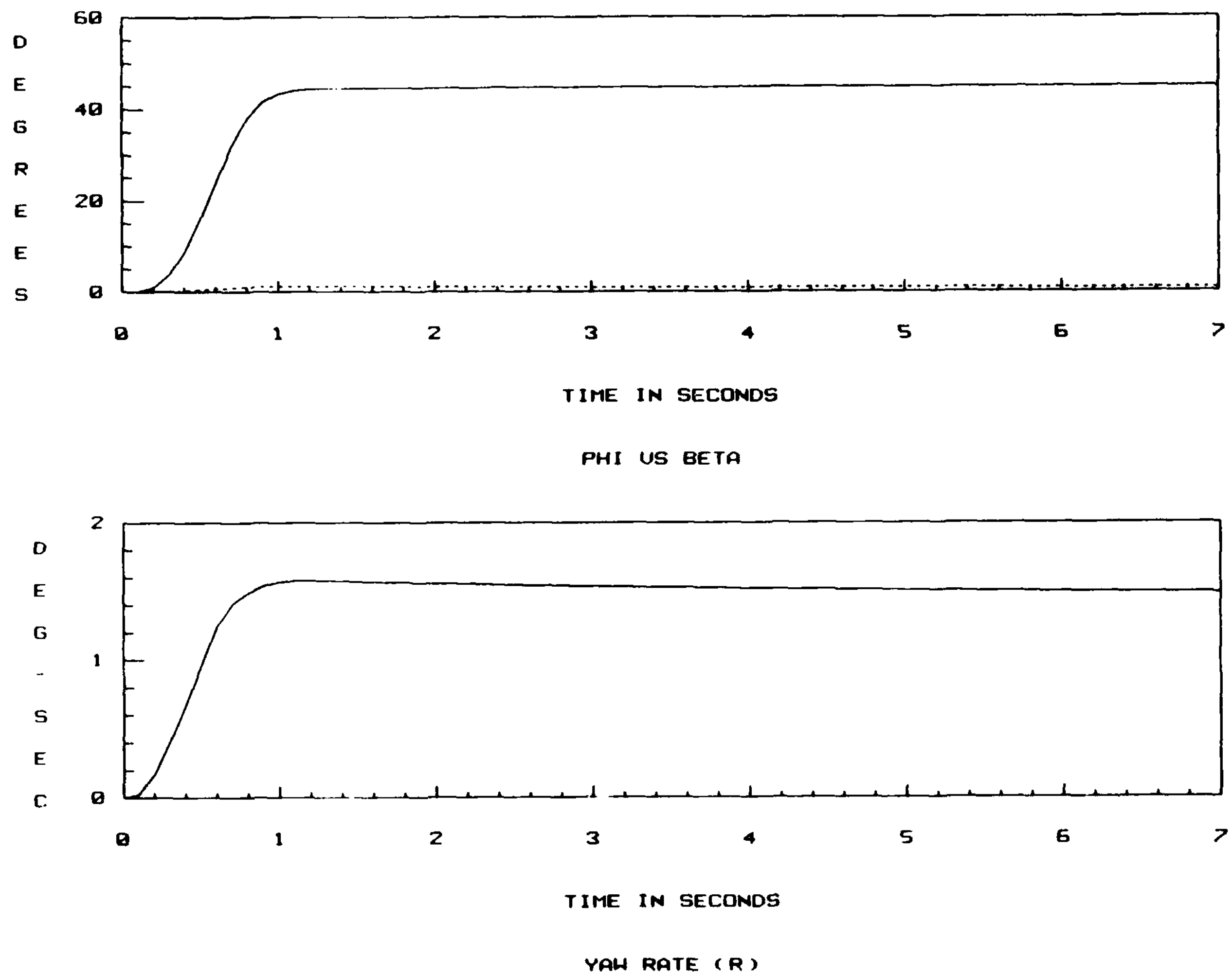


Figure E.2. ϕ, β, and r - 45° Banked Turn (Continuous)

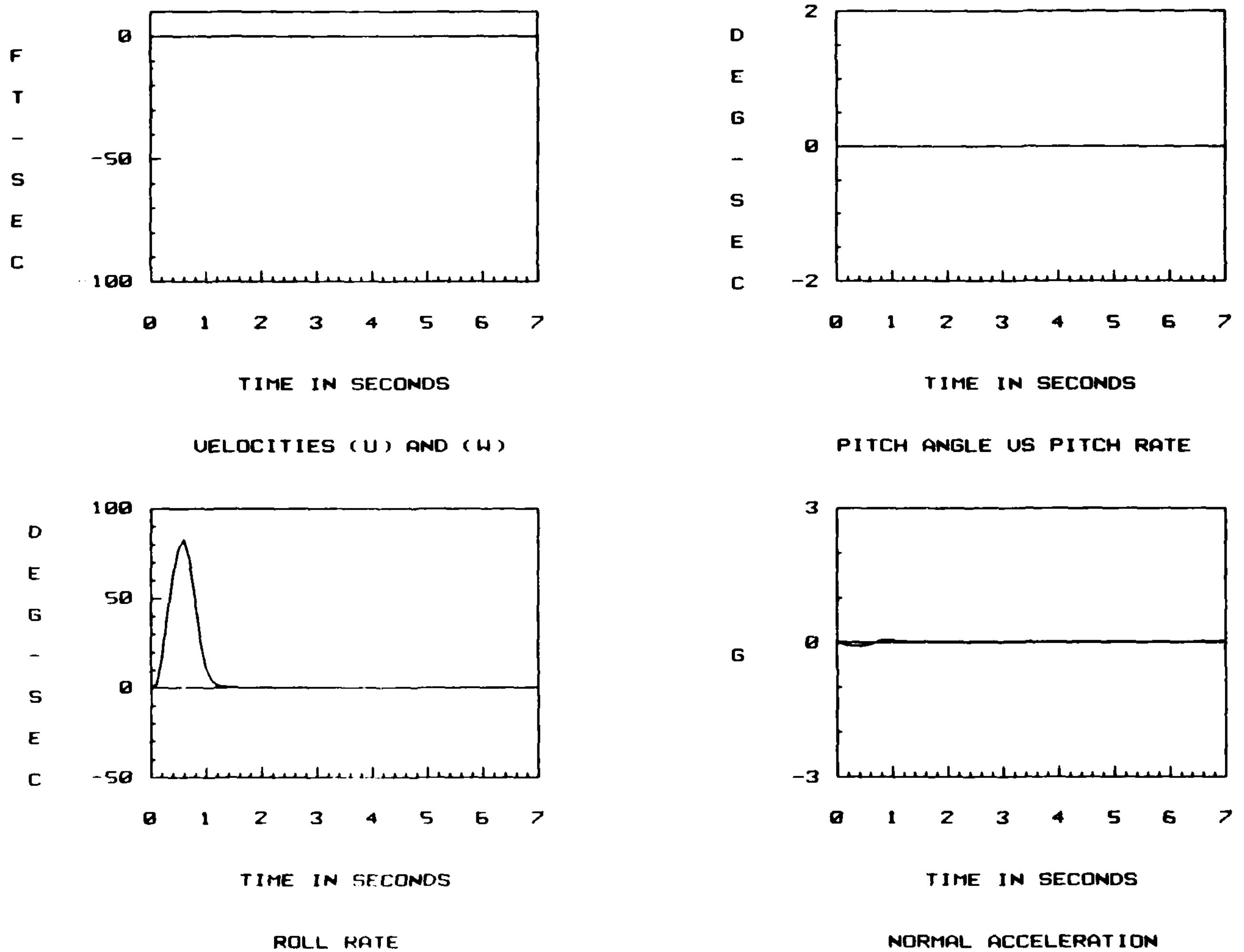


Figure E.3. u, w, θ, q, p, and Nz - 45° Banked Turn (Continuous)

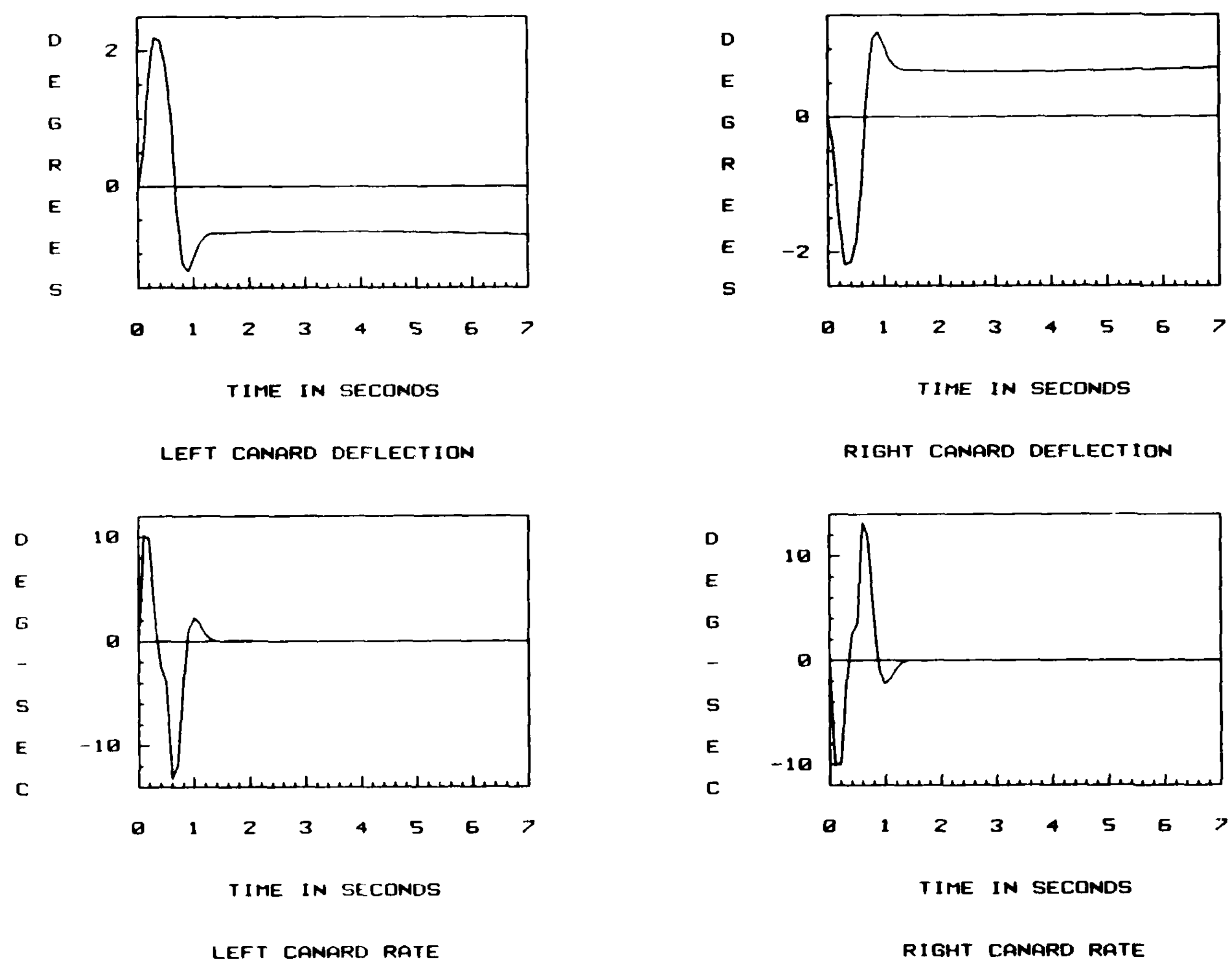


Figure E.4. Canard Deflection and Rates - 45° Banked Turn (Continuous)

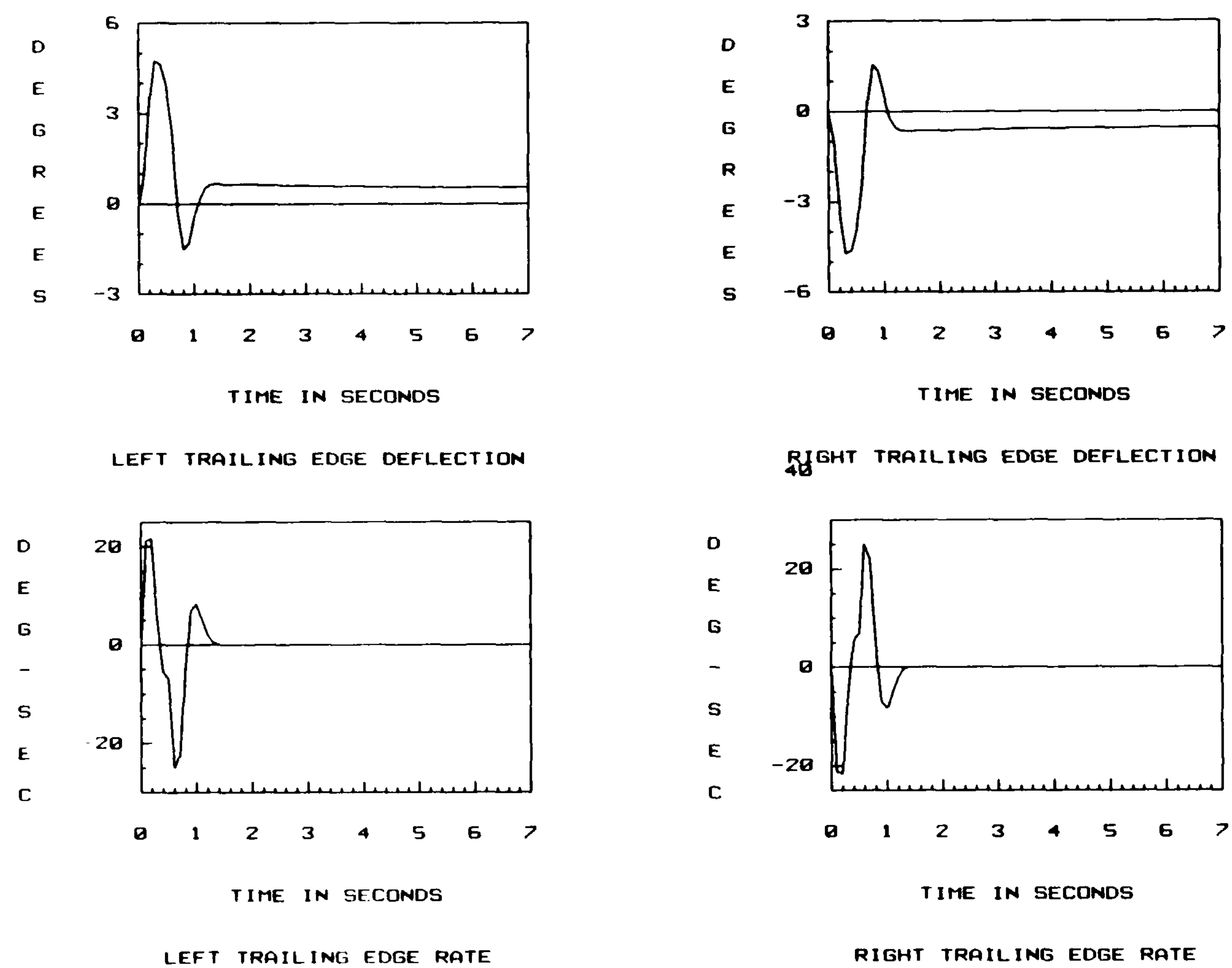


Figure E.5. Trailing Edge Deflection and Rates- 45° Banked Turn (Continuous)

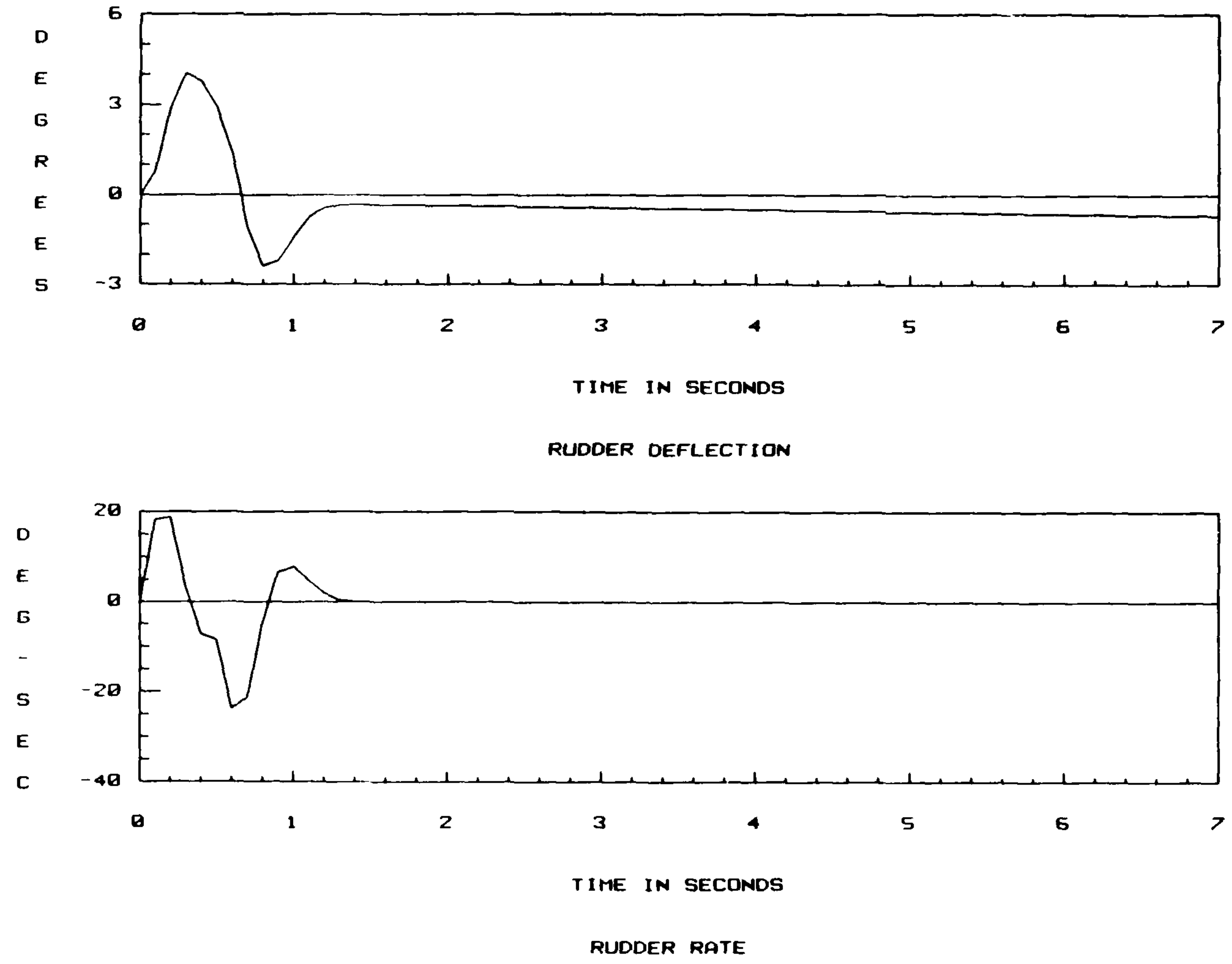


Figure E.6. Rudder Deflection and Rate - 45° Banked Turn (Continuous)

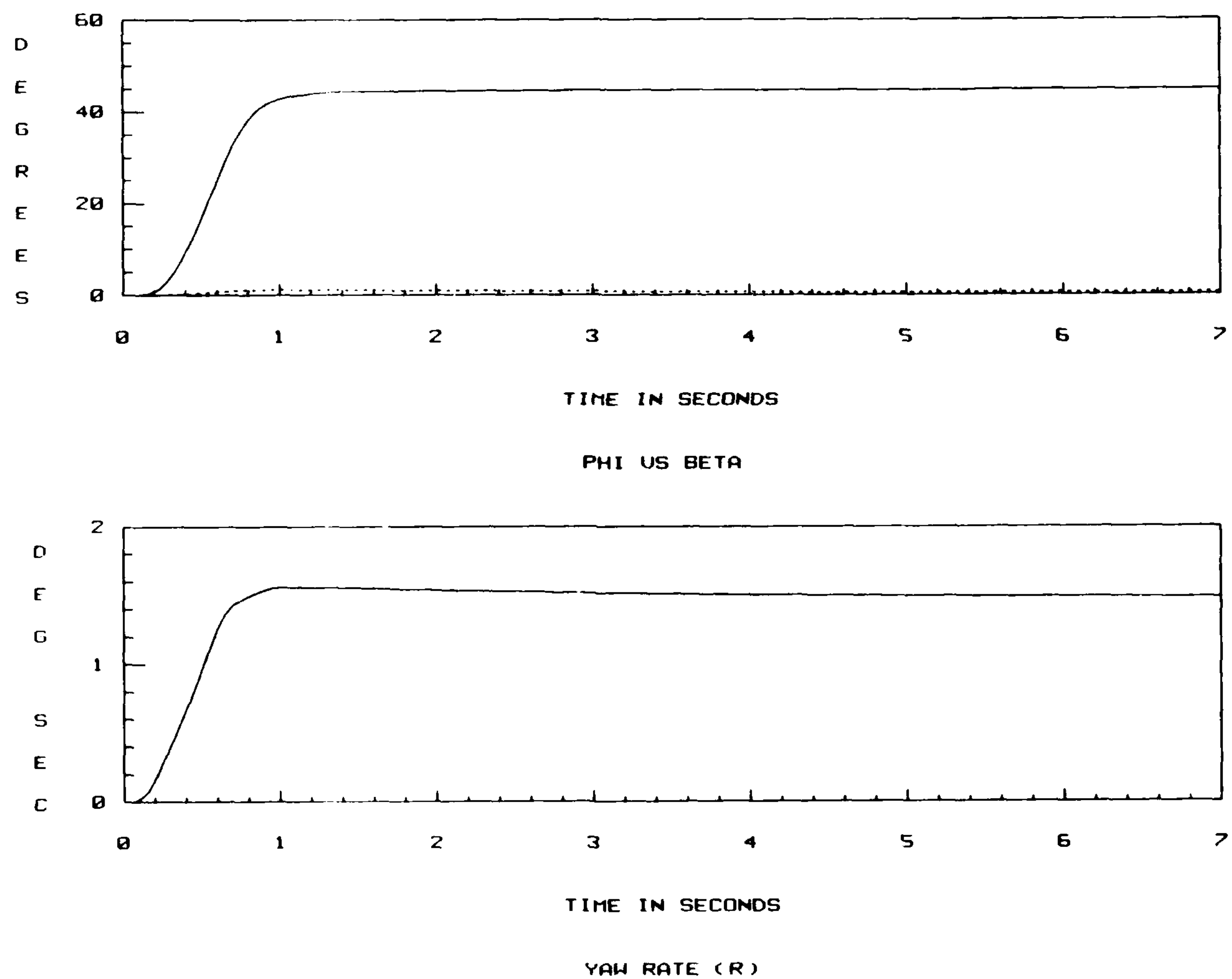


Figure E.7. ϕ, β, and r - 45° Banked Turn (Discrete)

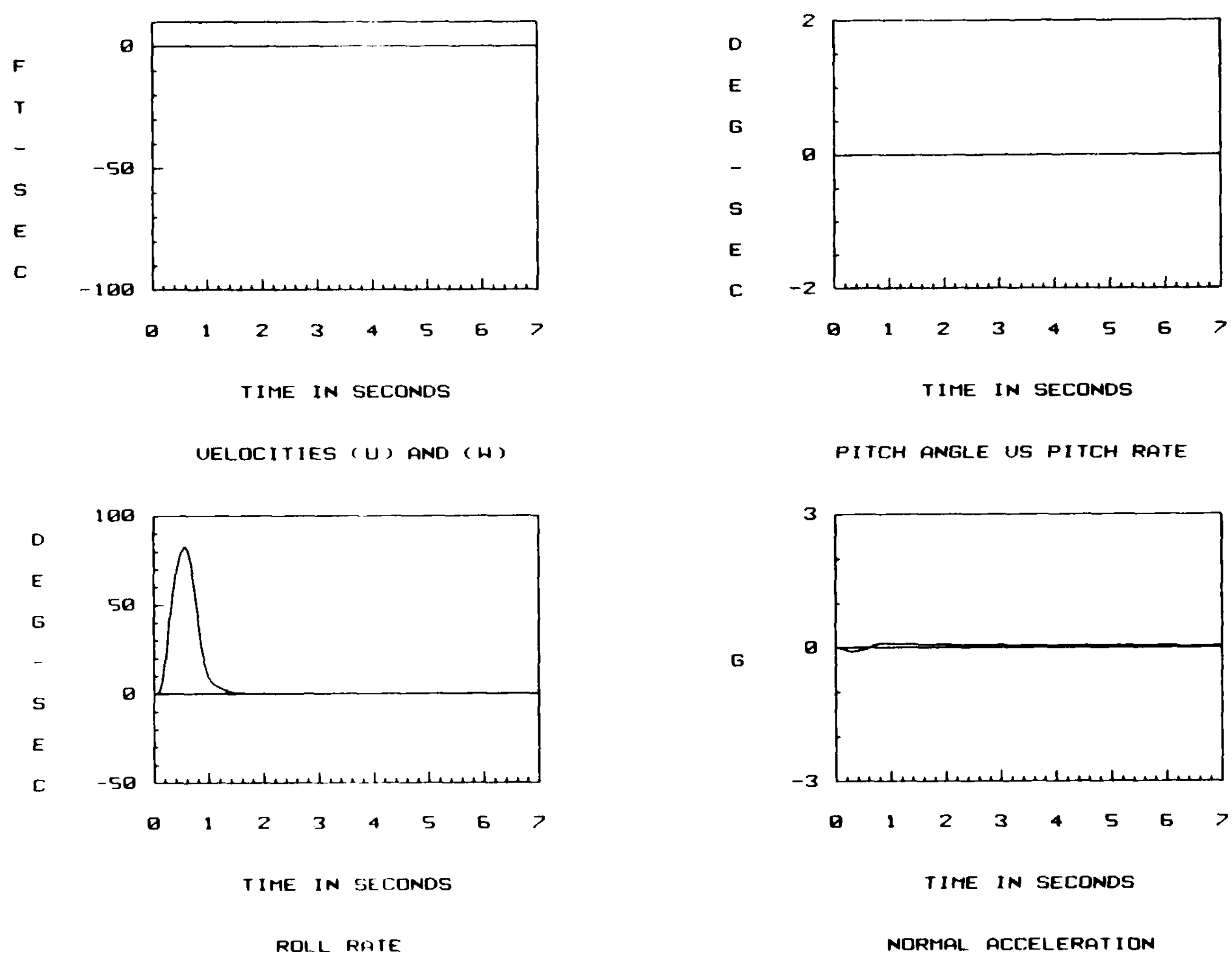


Figure E.8. u, w, θ, q, p, and Nz - 45° Banked Turn (Discrete)

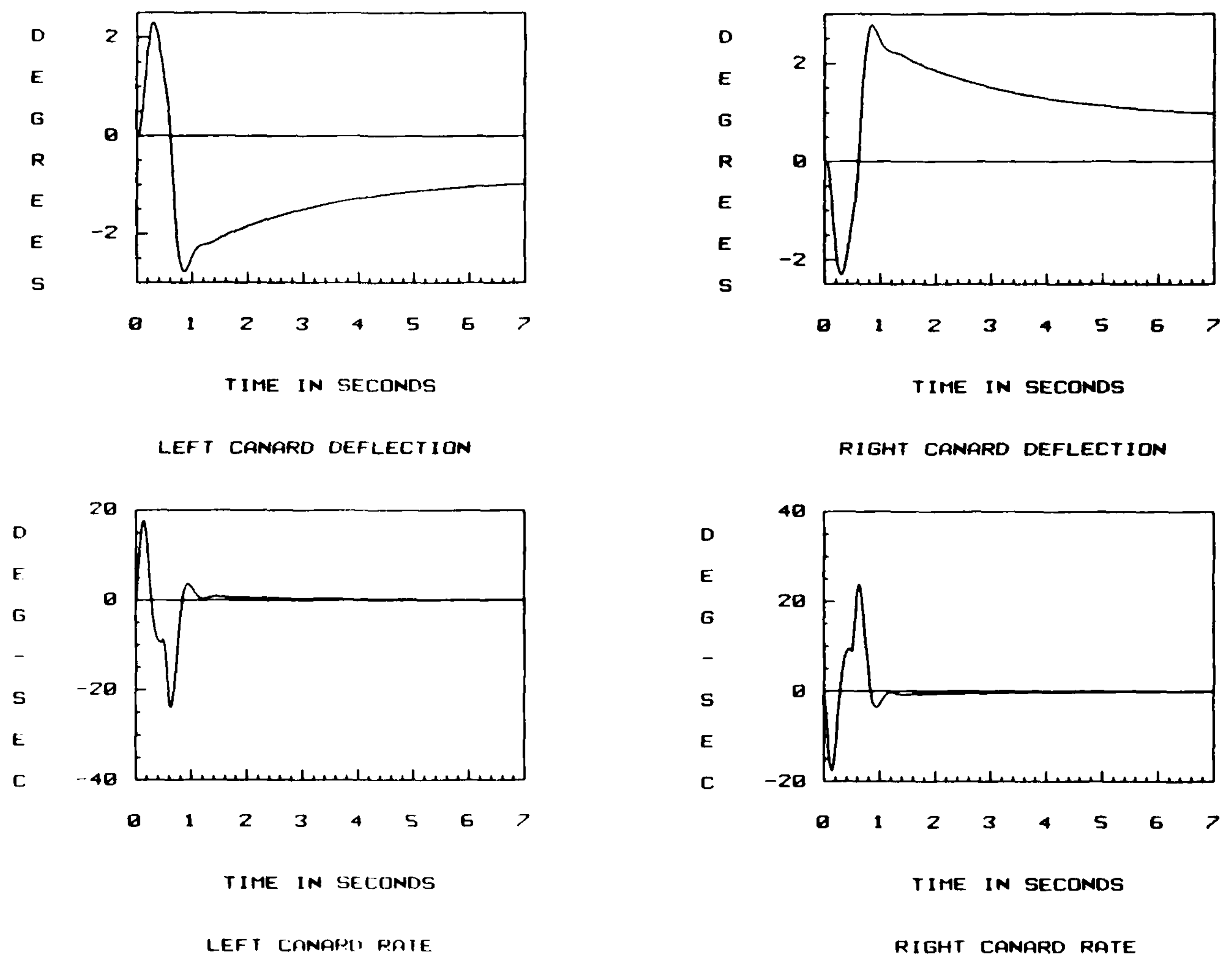


Figure E.9. Canard Deflection and Rates - 45° Banked Turn (Discrete)

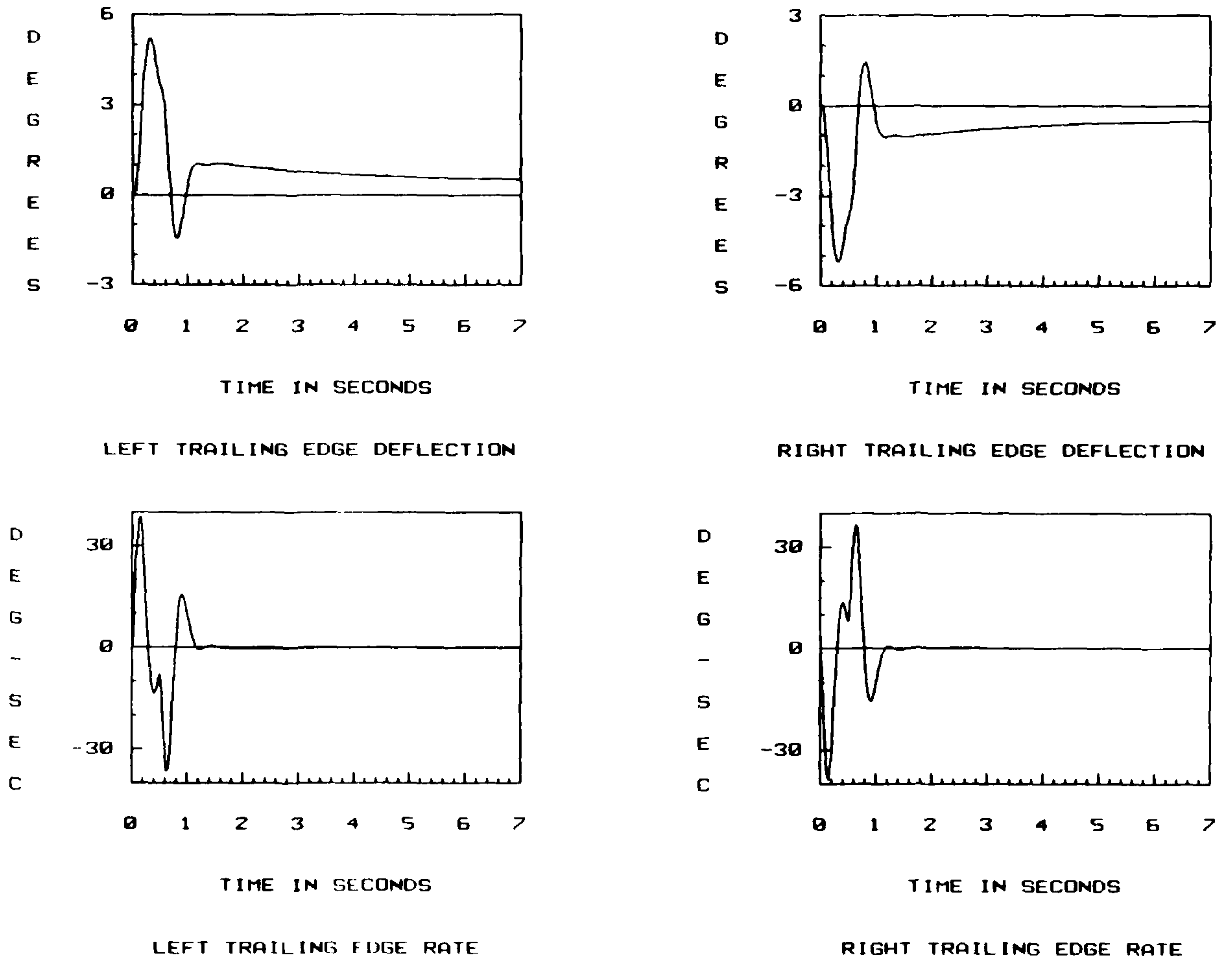


Figure E.10. Trailing Edge Deflection and Rates- 45° Banked Turn (Discrete)

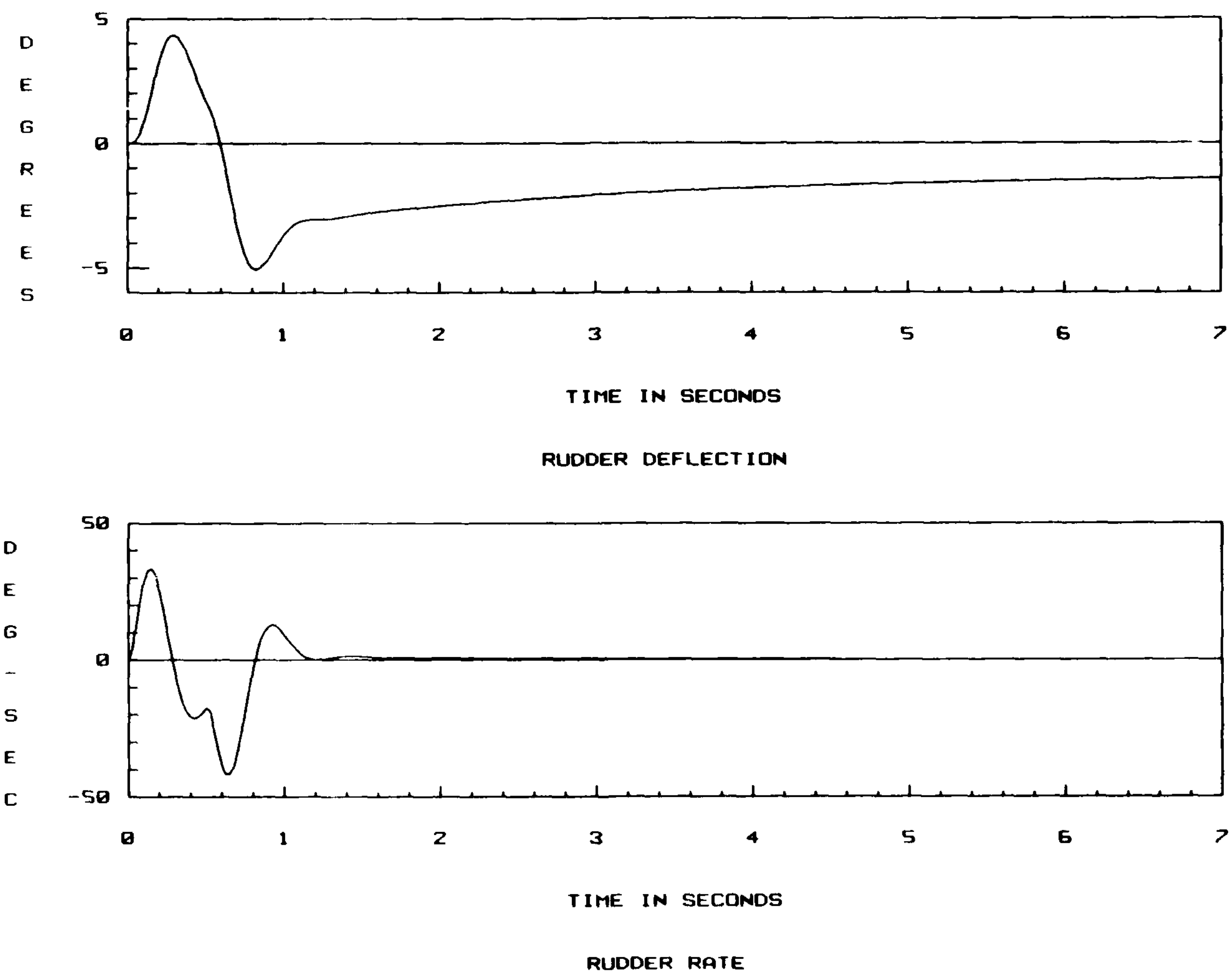
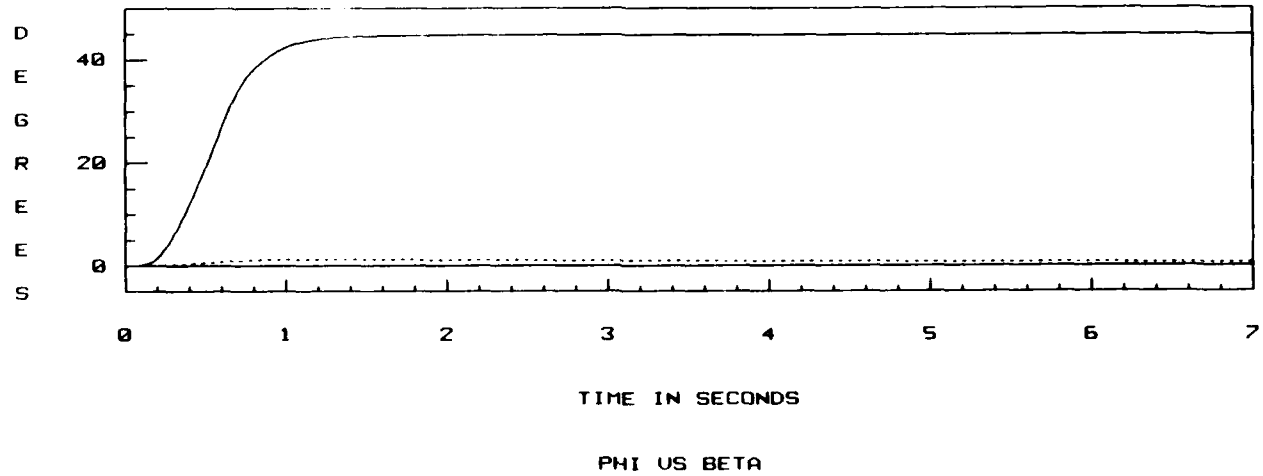


Figure E.11. Rudder Deflection and Rate - 45° Banked Turn (Discrete)



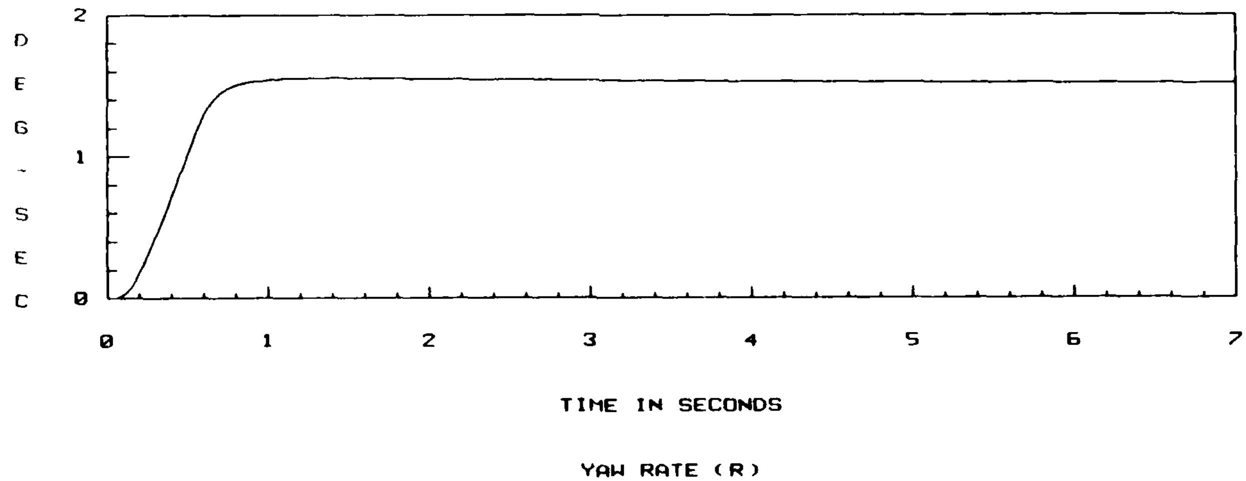


Figure E.12. ϕ, β, and r - 45° Banked Turn (Discrete using Step-Response Matrix)

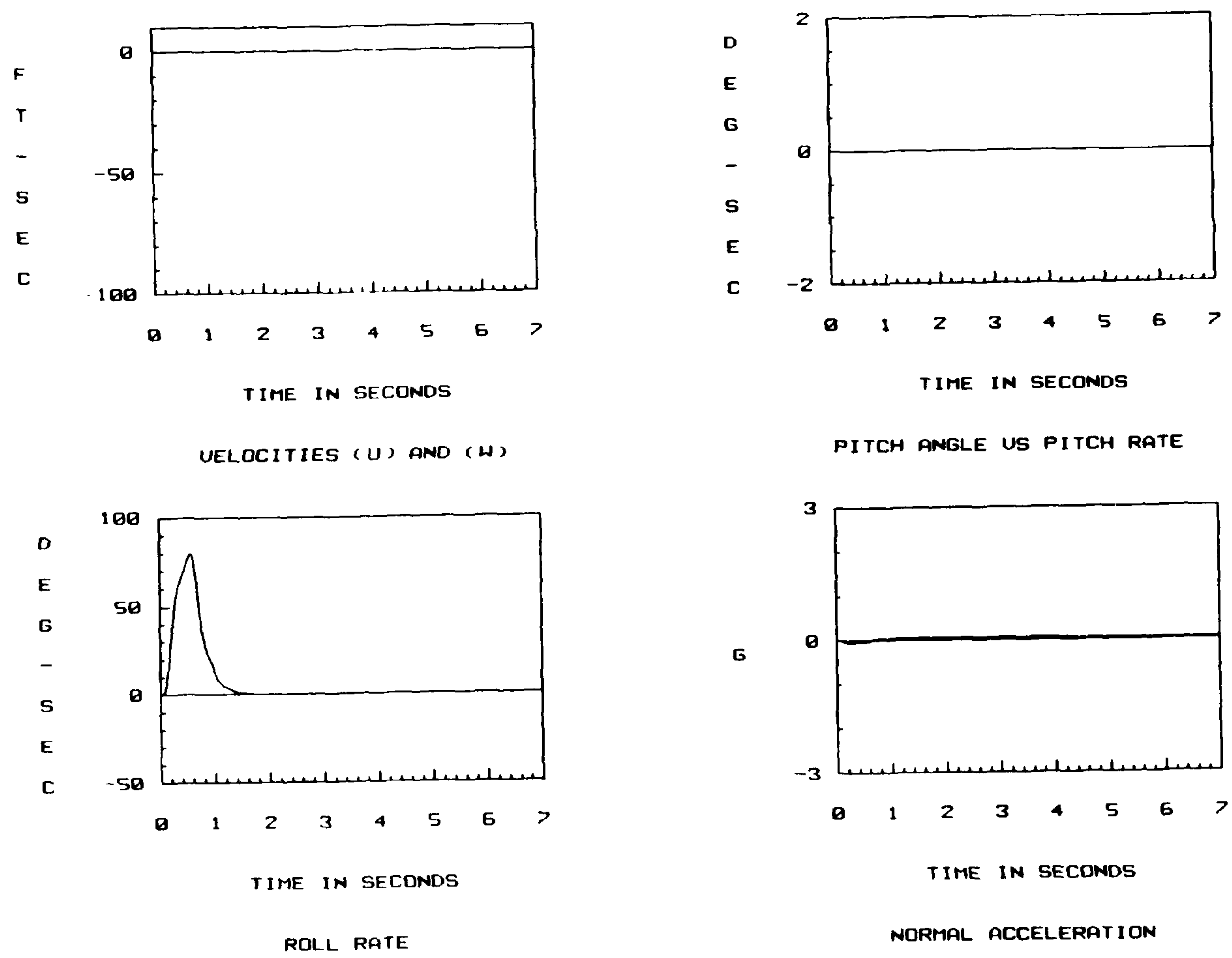


Figure E.13. u, w, θ, q, p, and Nz - 45° Banked Turn (Discrete using Step-Response Matrix)

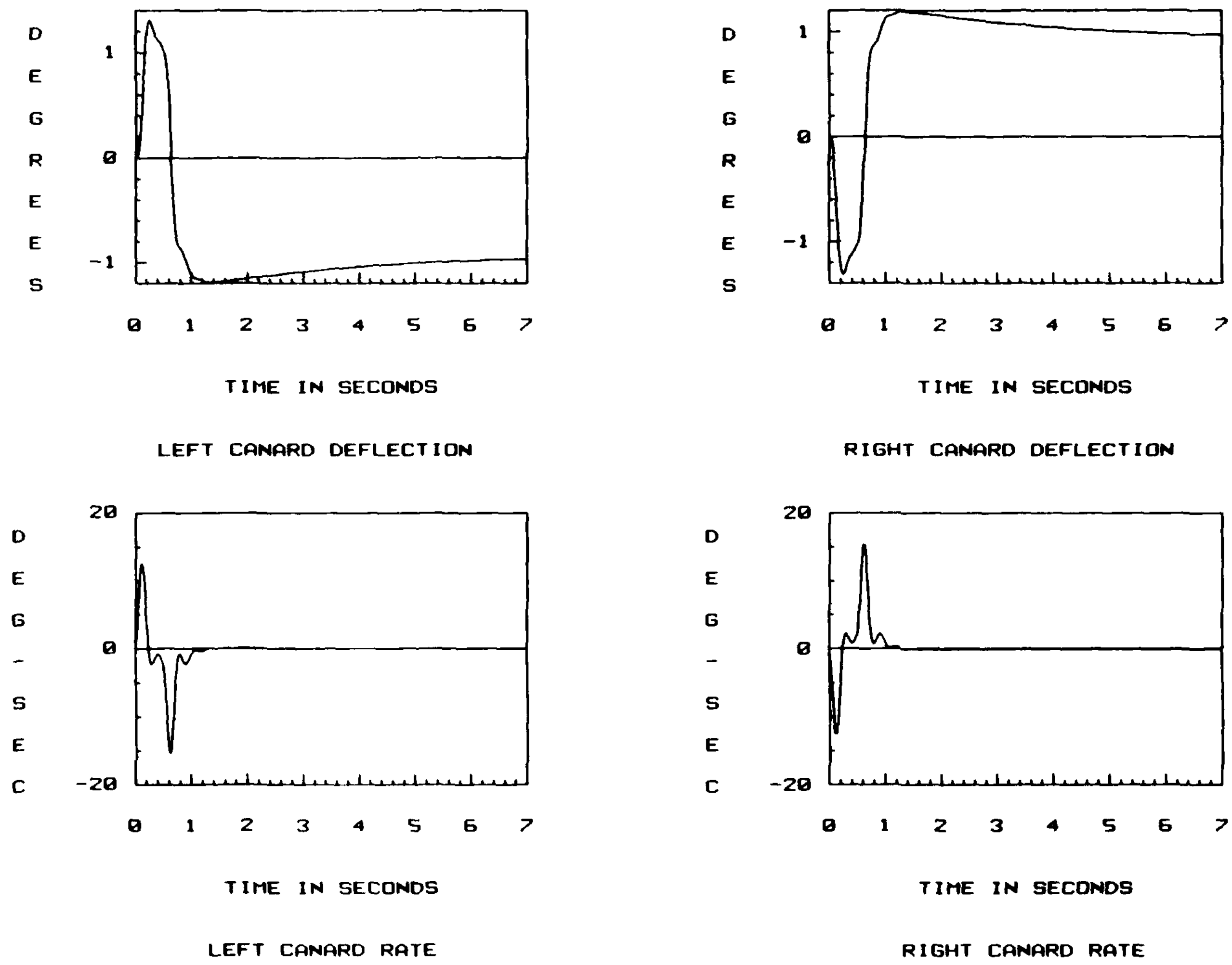


Figure E.14. Canard Deflection and Rates - 45° Banked Turn (Discrete using Step-Response Matrix)

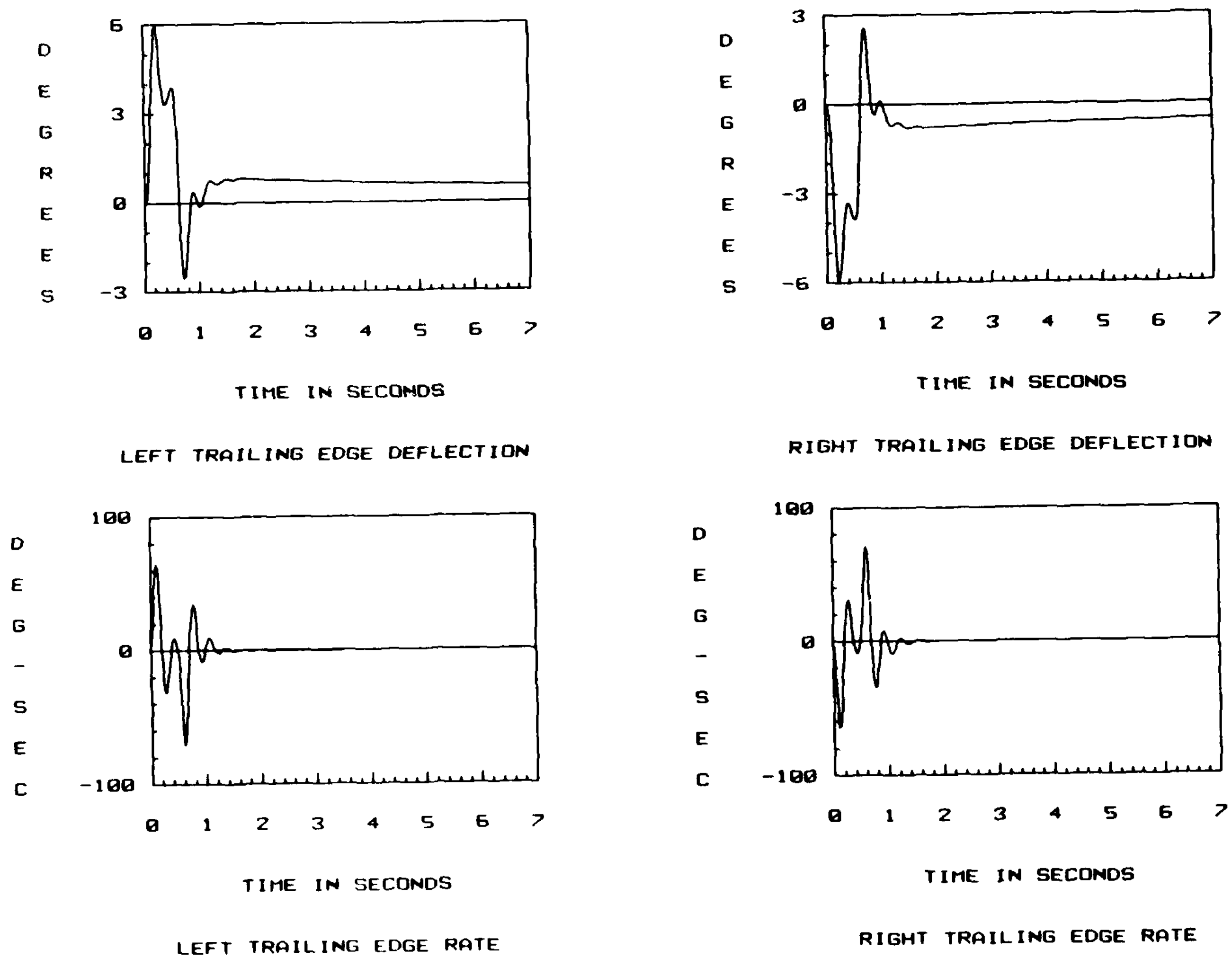


Figure E.15. Trailing Edge Deflection and Rates- 45° Banked Turn (Discrete using Step-Response Matrix)

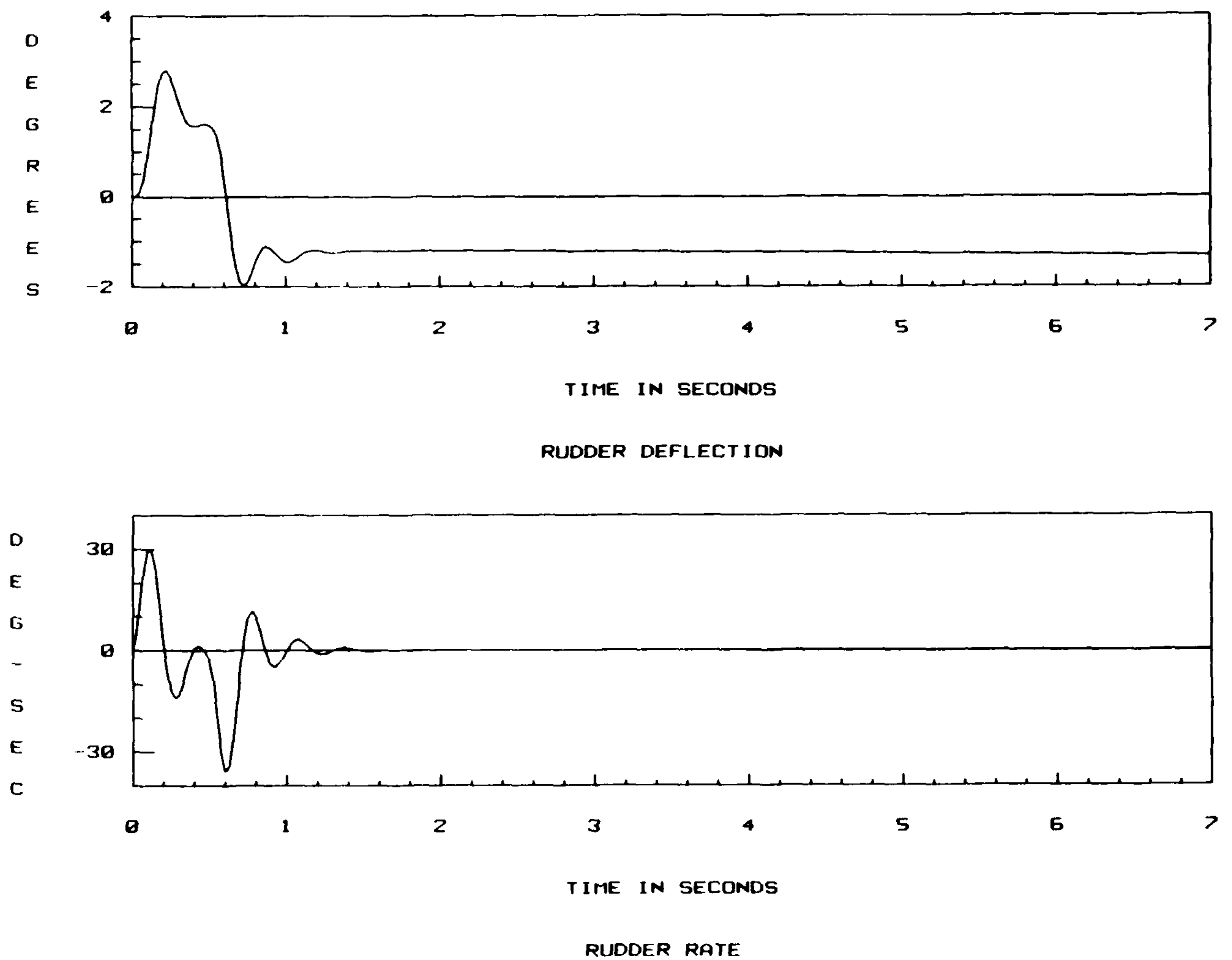


Figure E.16. Rudder Deflection and Rate - 45° Banked Turn (Discrete using Step-Response Matrix)

Responses for the pitch tracking command are show starting with Figure E.17 and illustrate the performance of the controller.

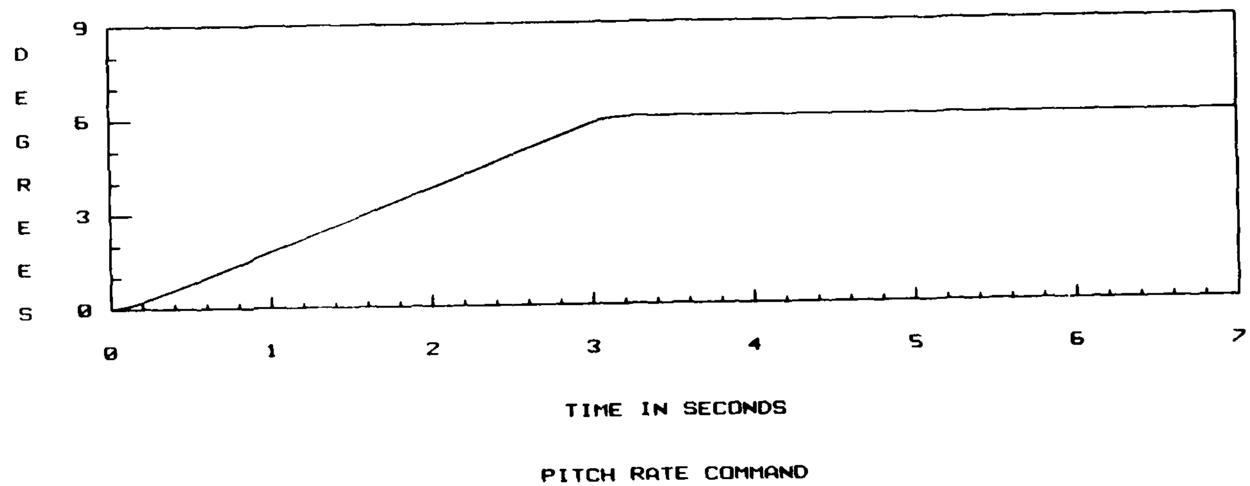
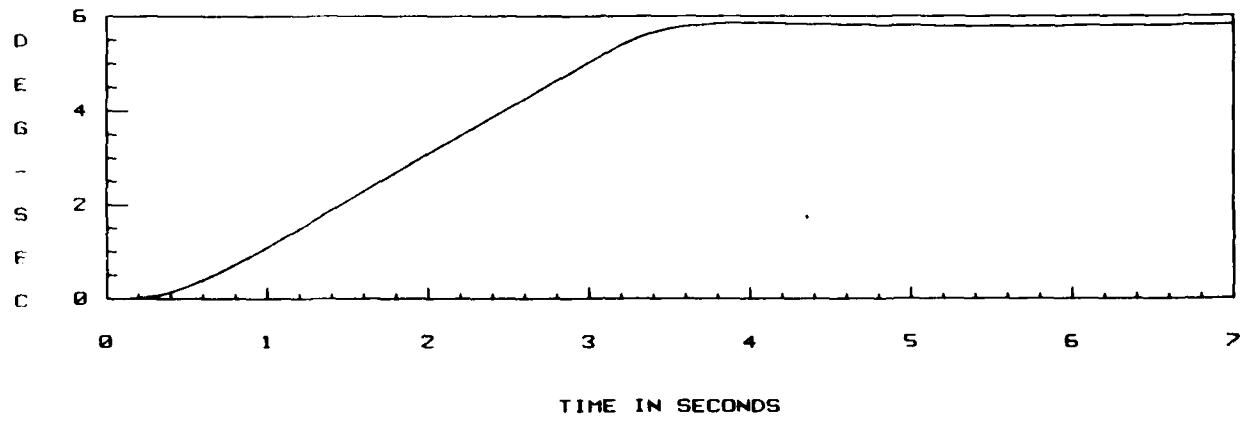
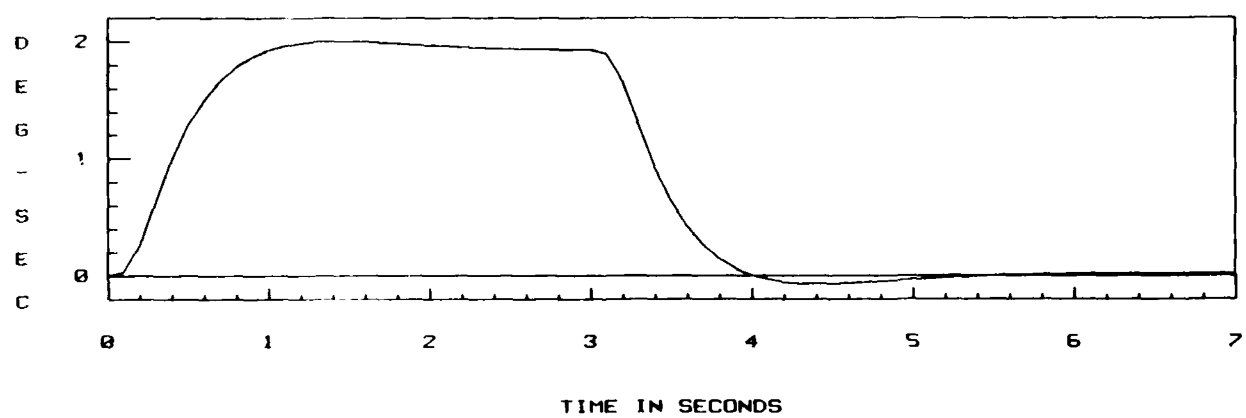


Figure E.17. θ_{cmd}



PITCH ANGLE



PITCH RATE

Figure E.18. θ and q - θ_{cmd} (Continuous)

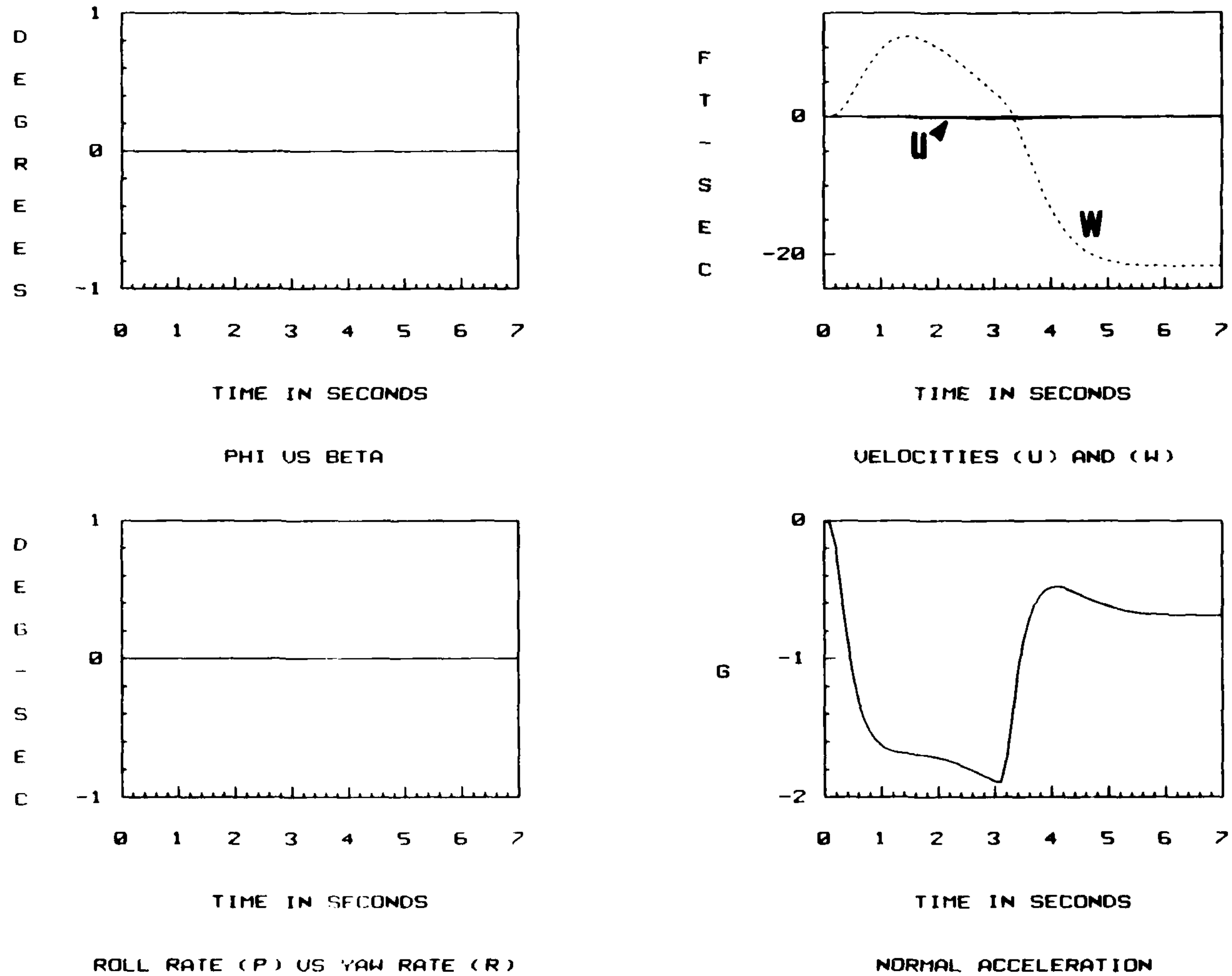


Figure E.19. ϕ, β, u, w, r, p, and Nz - θ_{cmd} (Continuous)

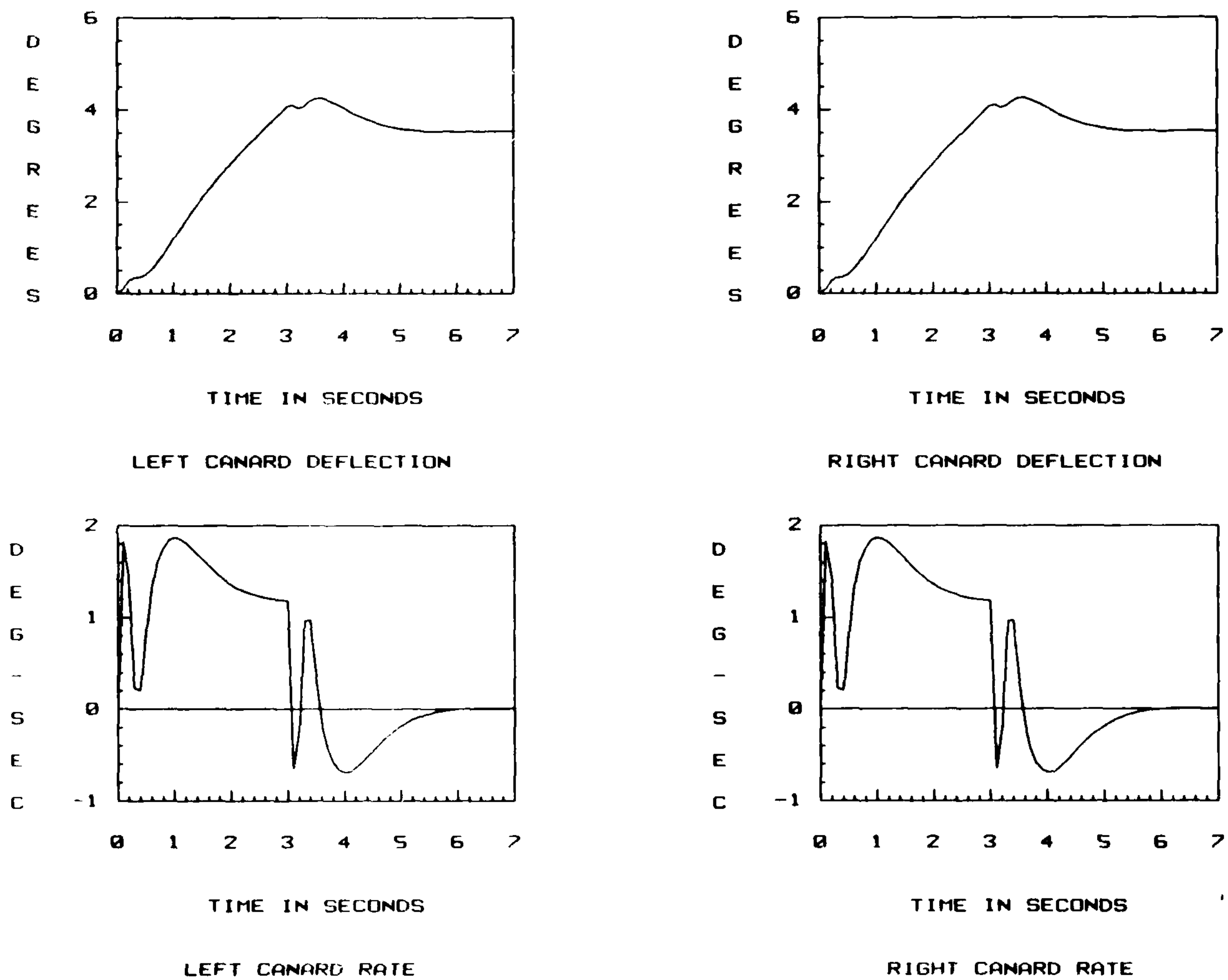
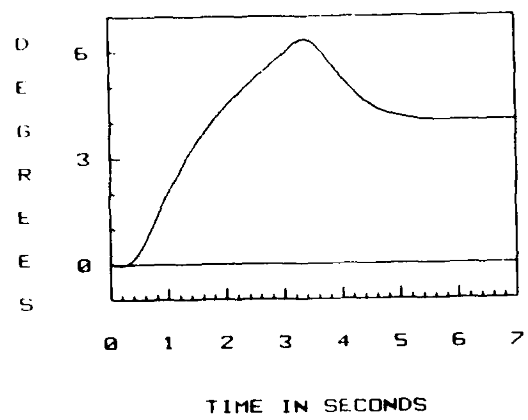
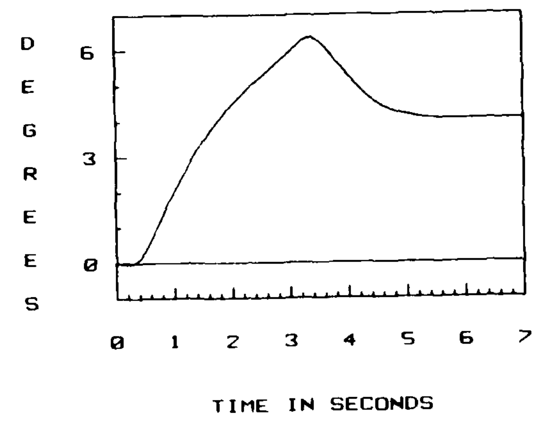


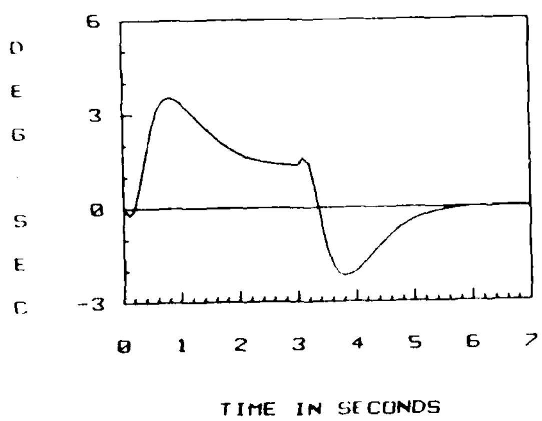
Figure E.20. Canard Deflection and Rates - θ_{cmd} (Continuous)



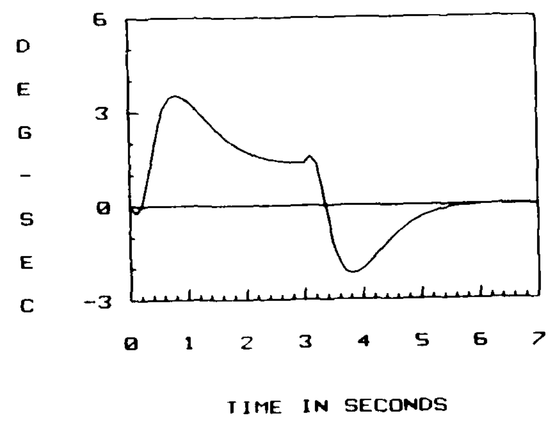
LEFT TRAILING EDGE DEFLECTION



RIGHT TRAILING EDGE DEFLECTION



LEFT TRAILING EDGE RATE



RIGHT TRAILING EDGE RATE

Figure E.21. Trailing Edge Deflection and Rates - θ_{cmd} (Continuous)

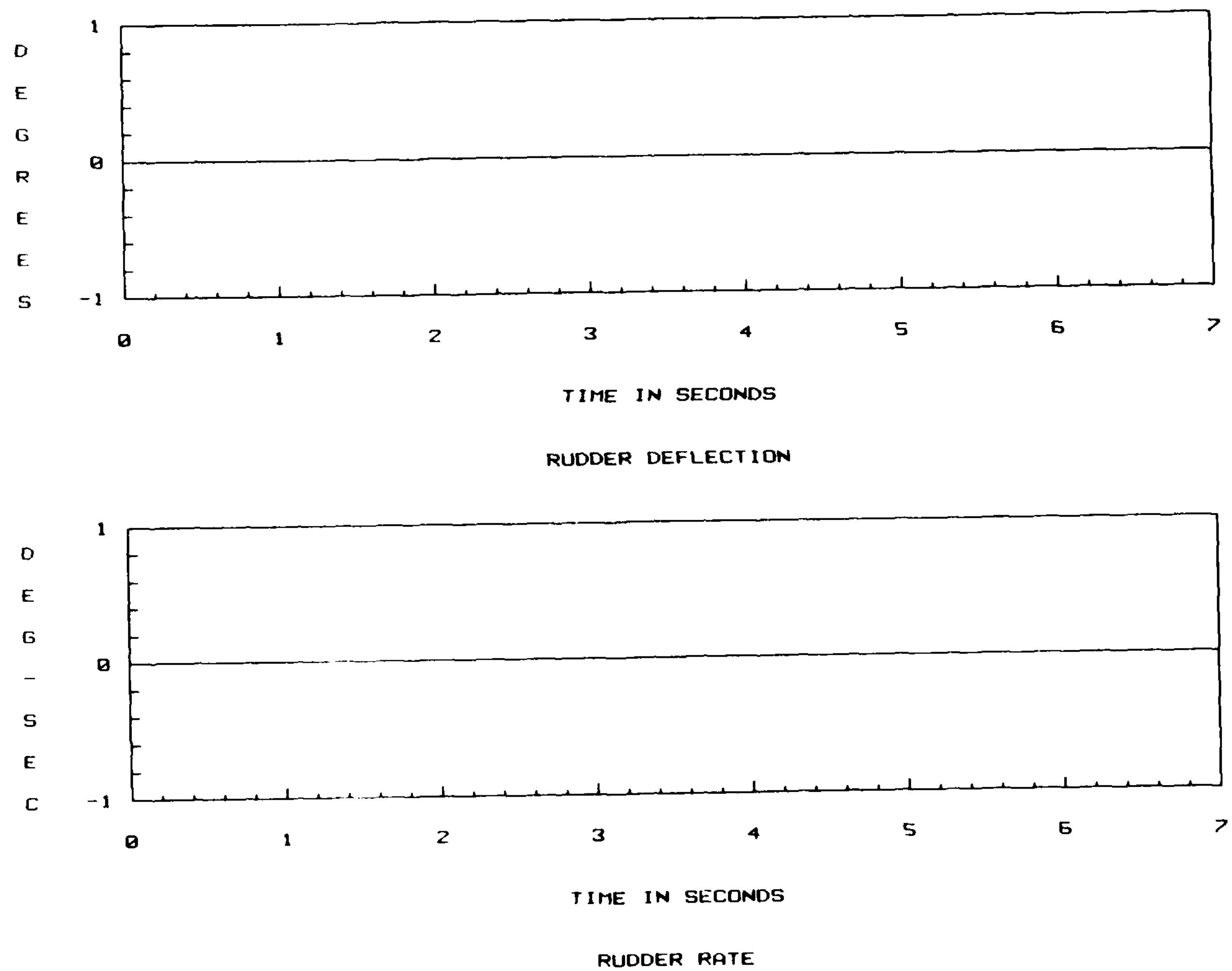


Figure E.22. Rudder Deflection and Rate - θ_{cmd} (Continuous)

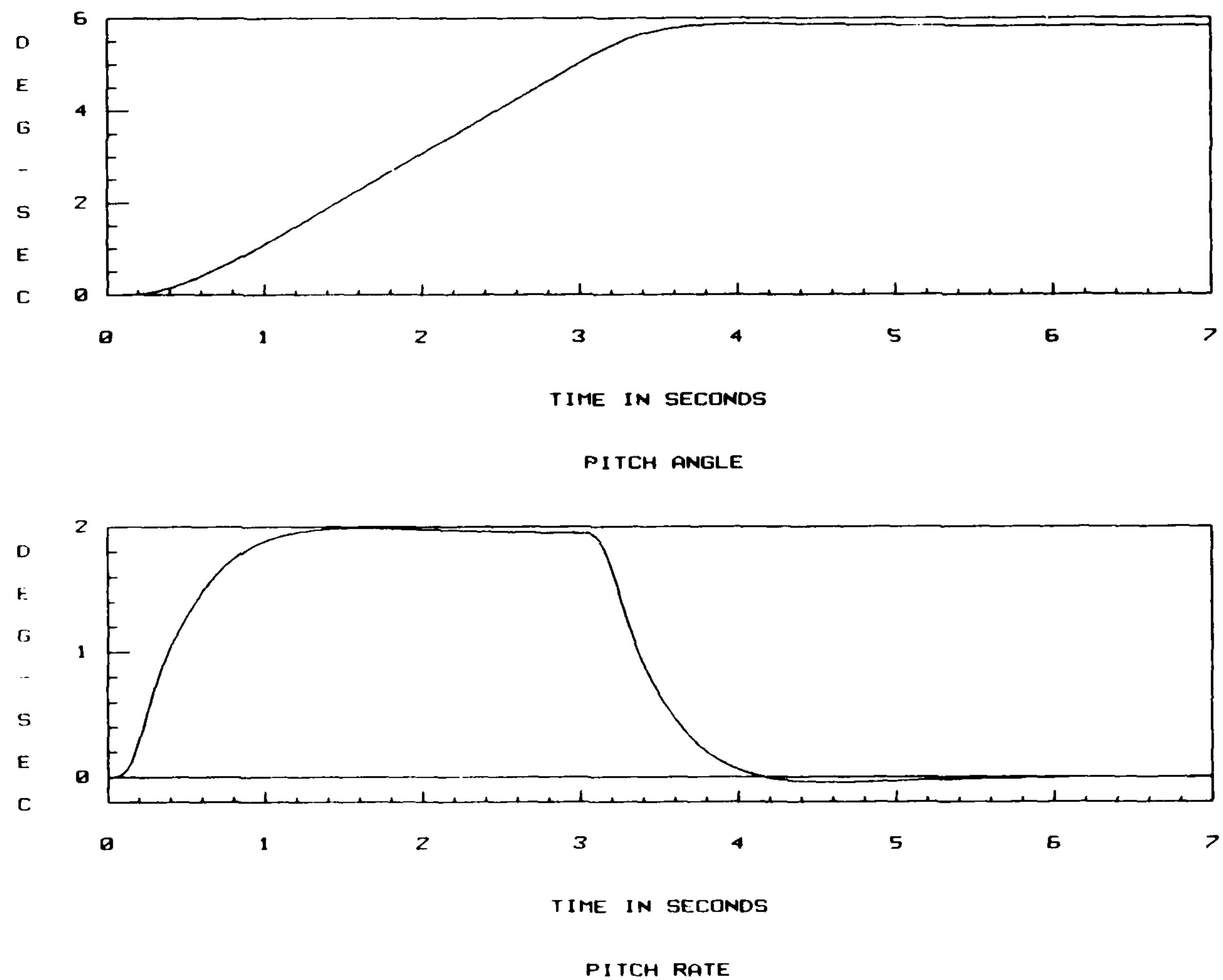


Figure E.23. θ and q - θ_{cmd} (Discrete)

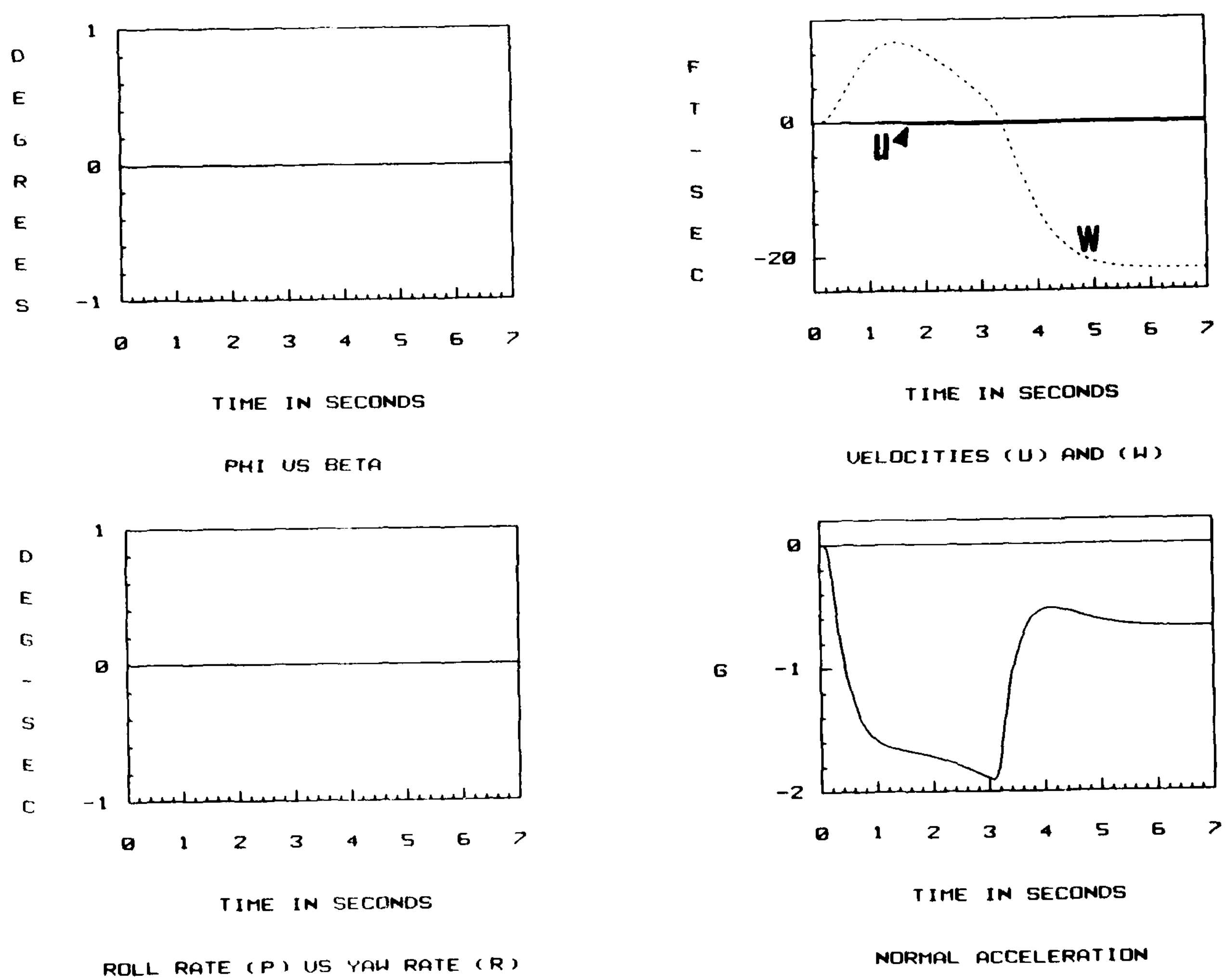


Figure E.24. ϕ, β, u, w, r, p, and Nz - θ_{cmd} (Discrete)

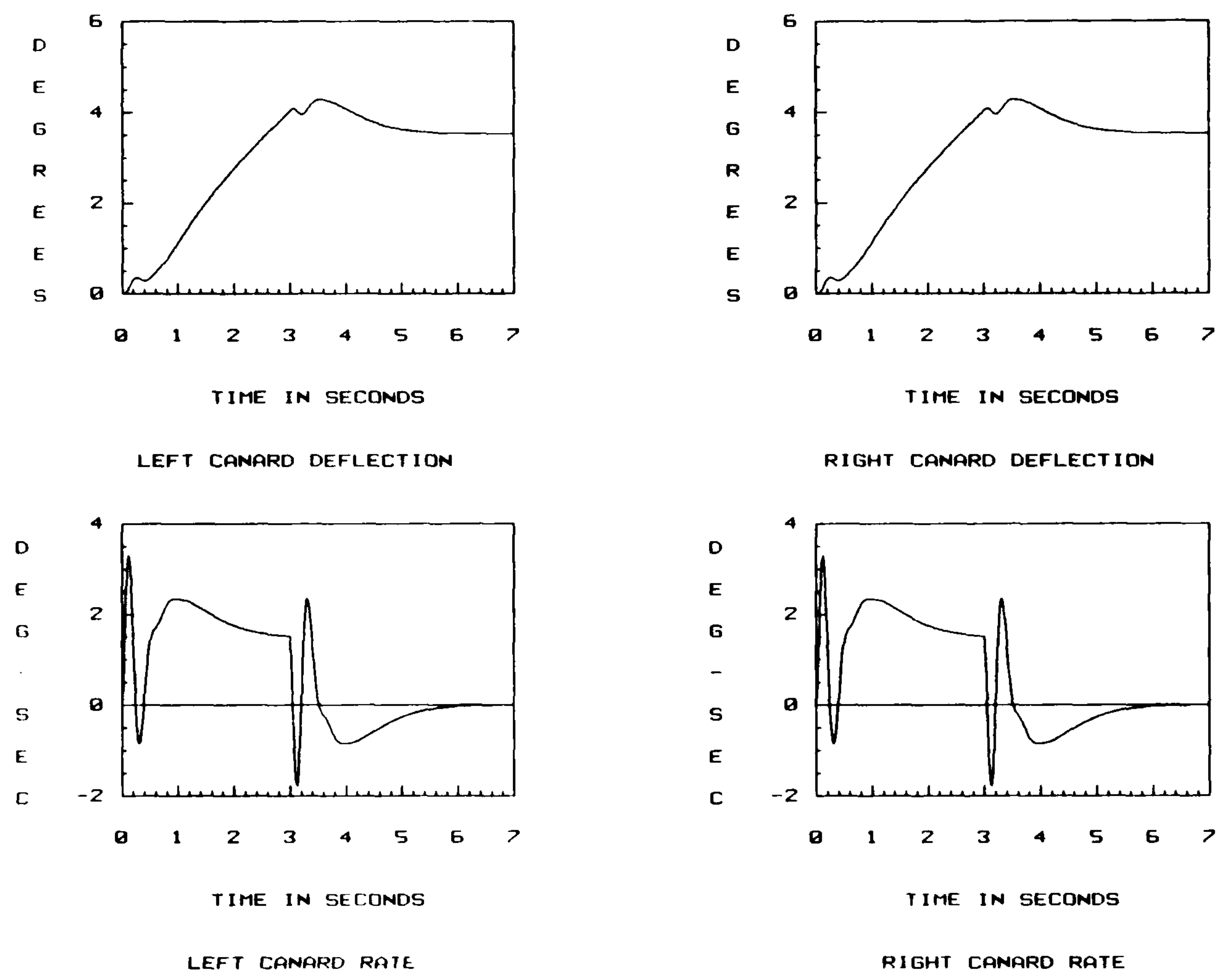


Figure E.25. Canard Deflection and Rates - θ_{cmd} (Discrete)

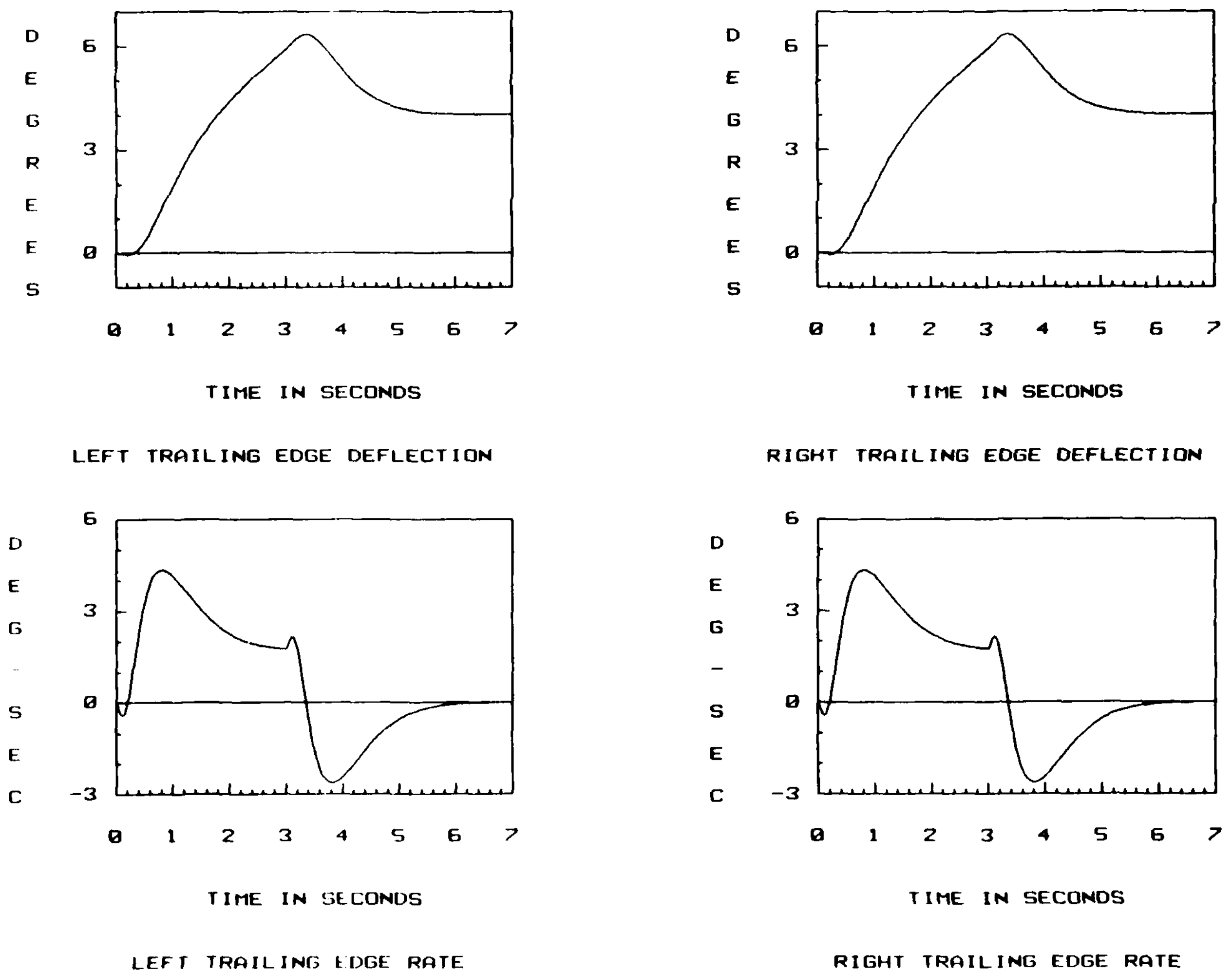
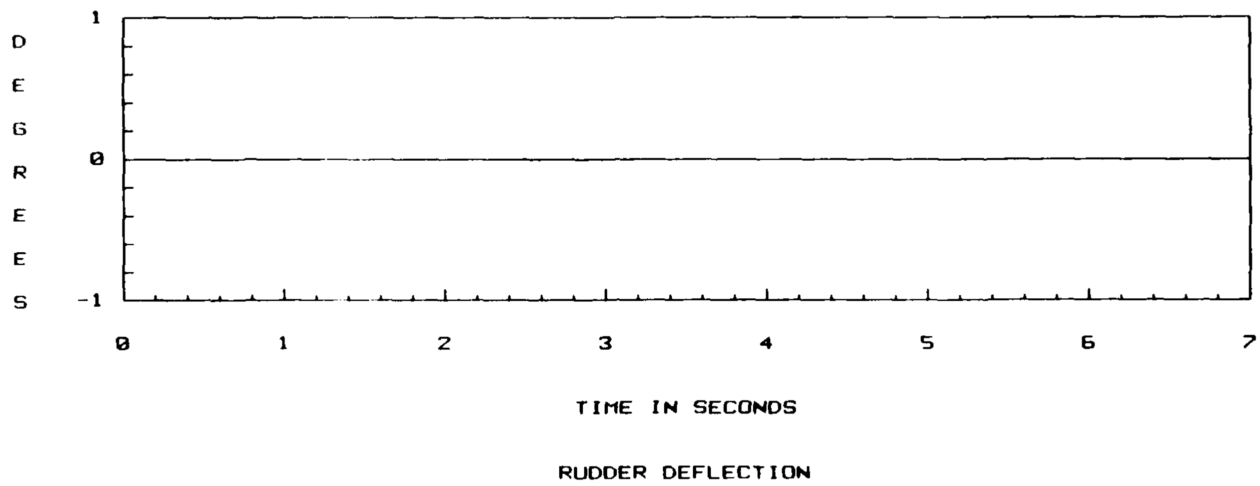


Figure E.26. Trailing Edge Deflection and Rates - θ_{cmd} (Discrete)



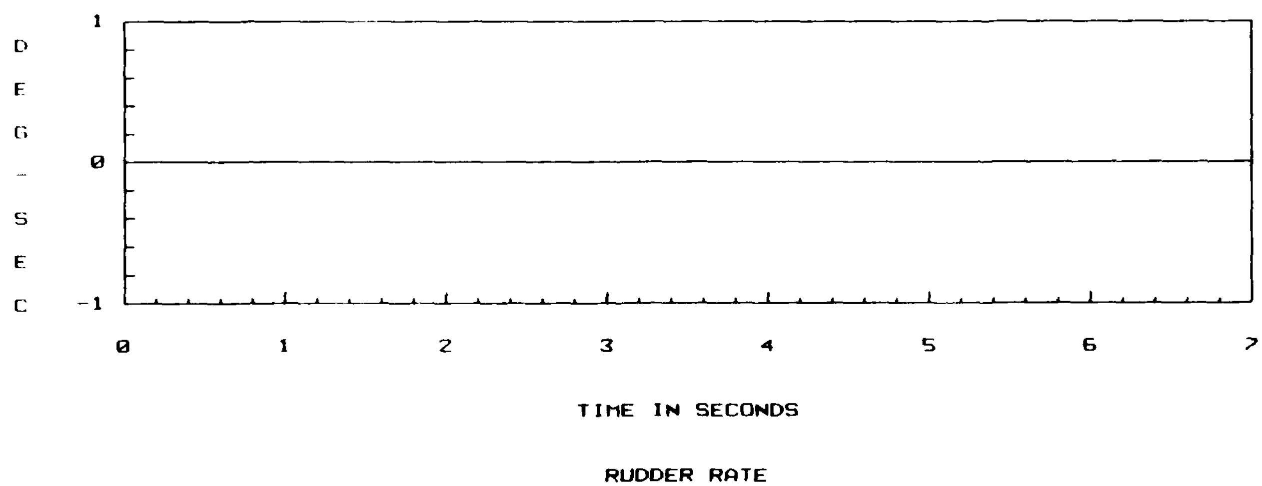
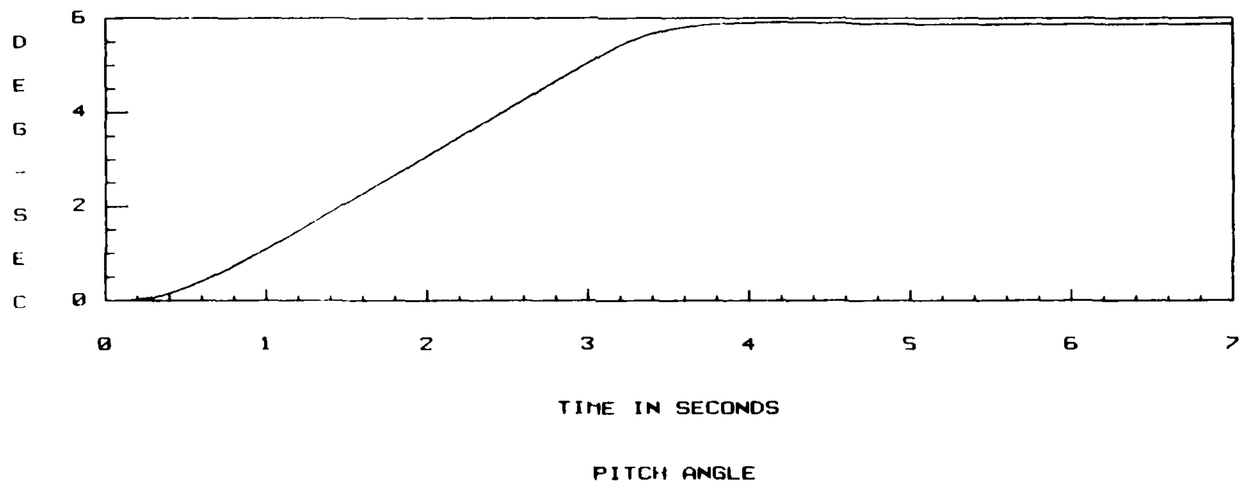


Figure E.27. Rudder Deflection and Rate - θ_{cmd} (Discrete)



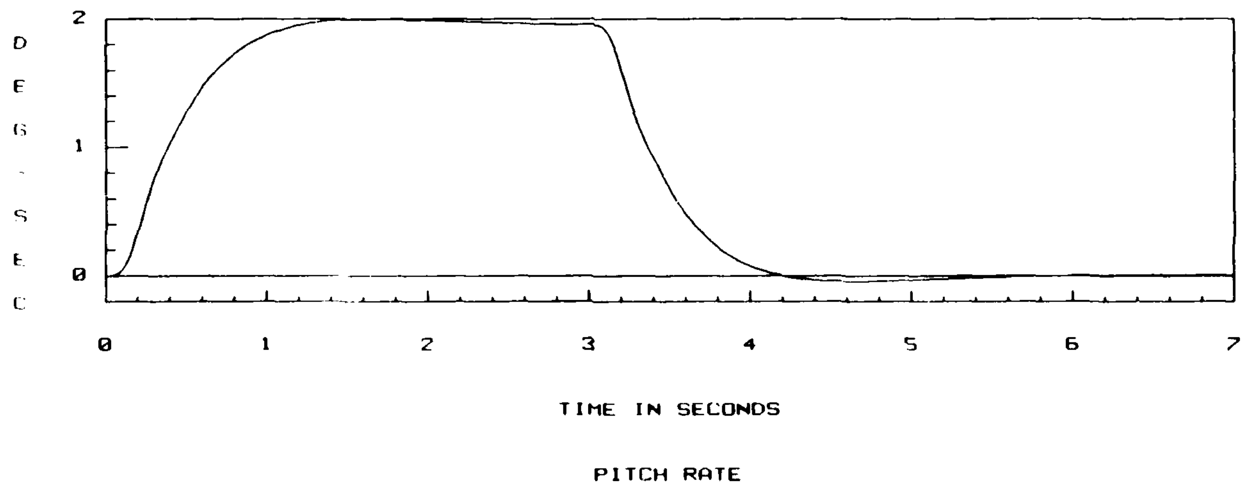


Figure E.28. θ and q - θ_{cmd} (Discrete using Step-Response Matrix)

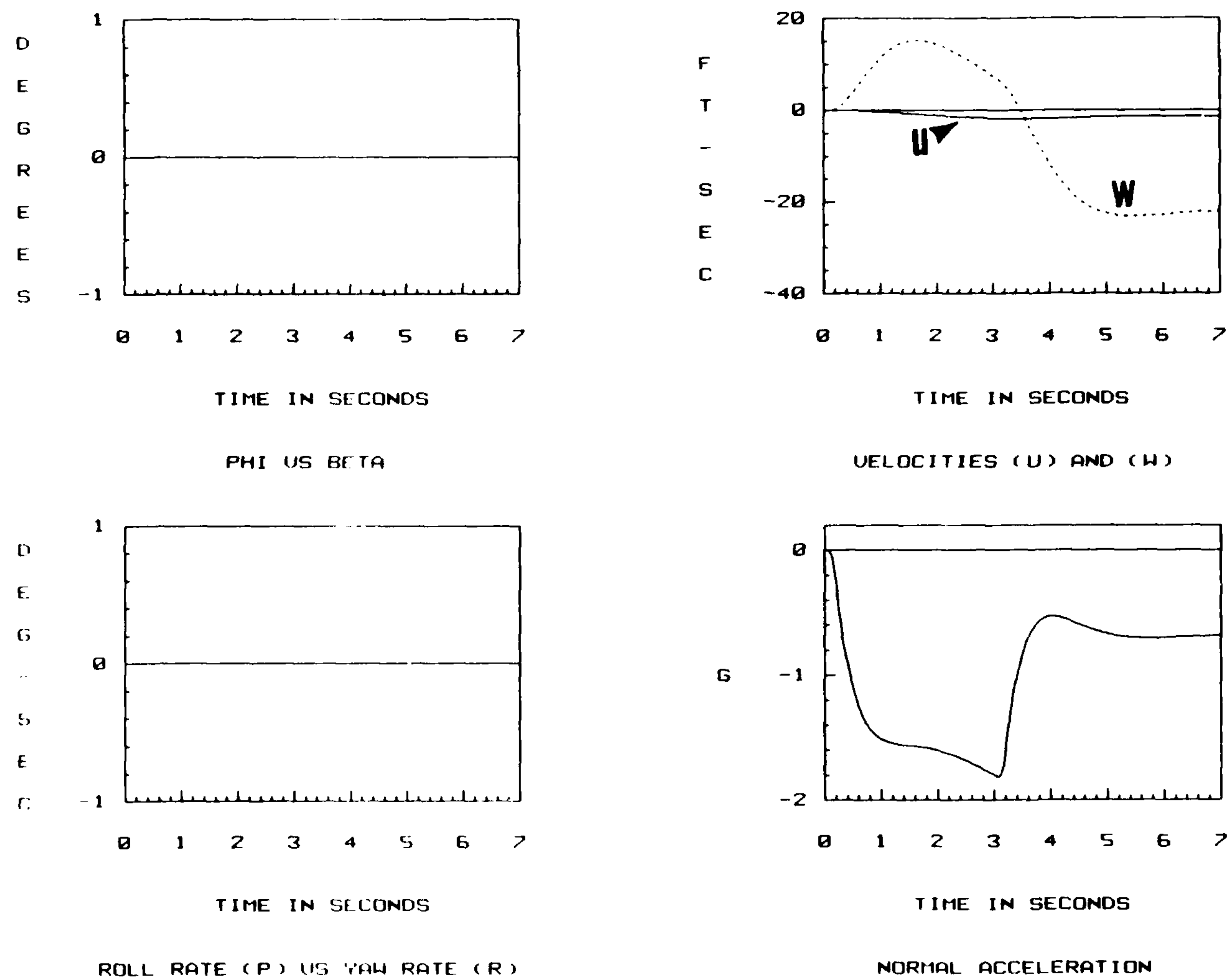


Figure E.29. ϕ, β, u, w, r, p, and Nz - θ_{cmd} (Discrete using Step-Response Matrix)

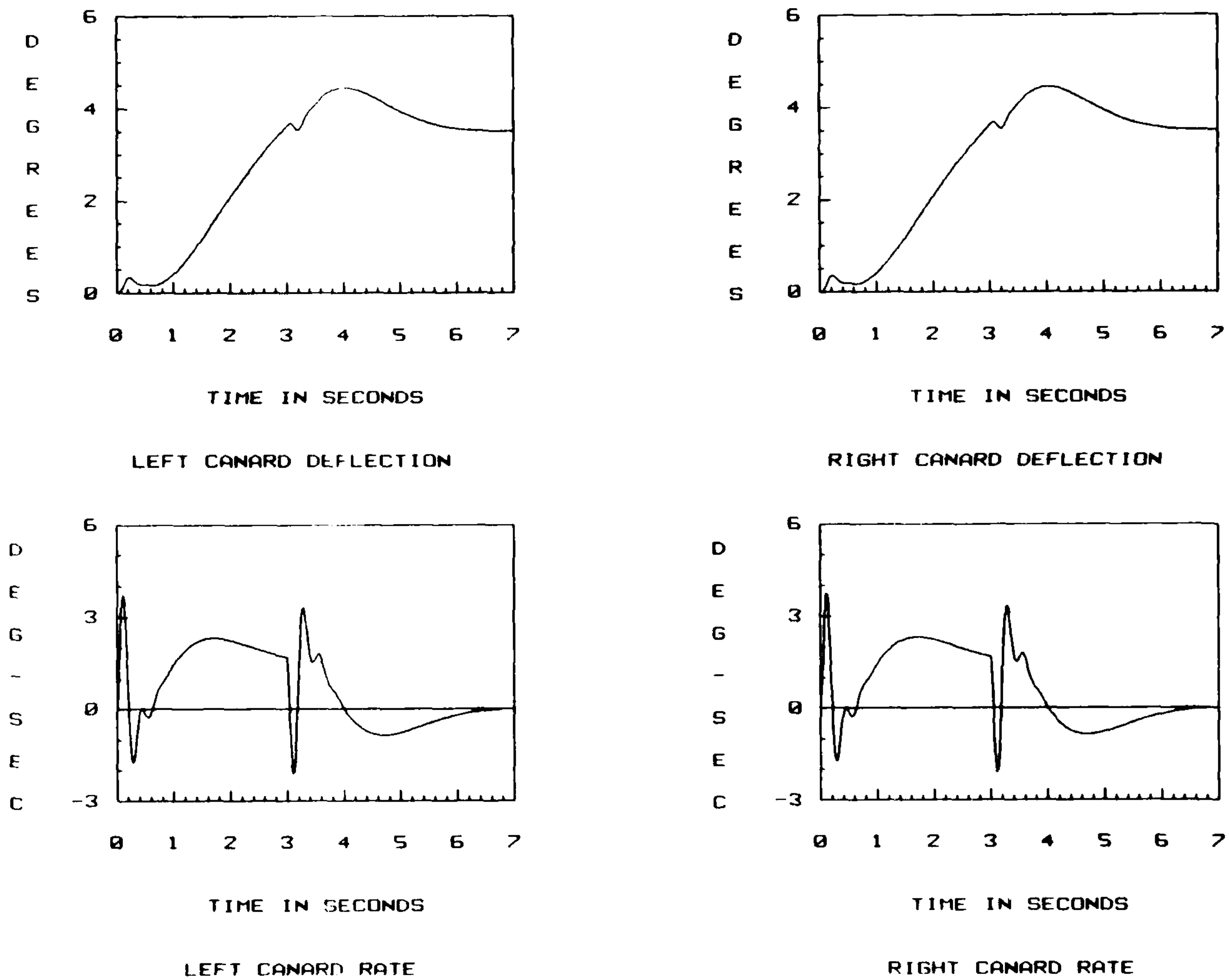


Figure E.30. Canard Deflection and Rates - θ_{cmd} (Discrete using Step-Response Matrix)

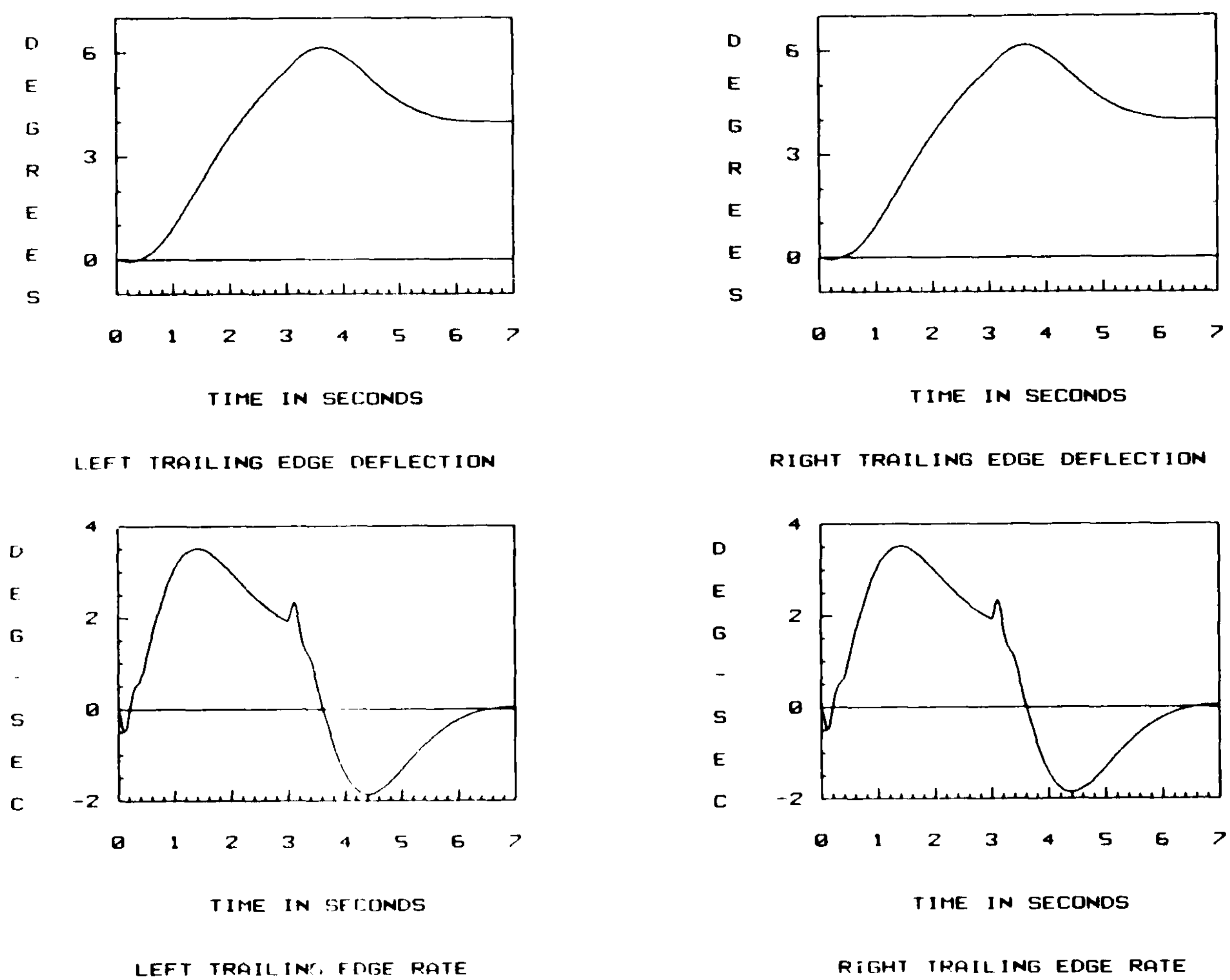
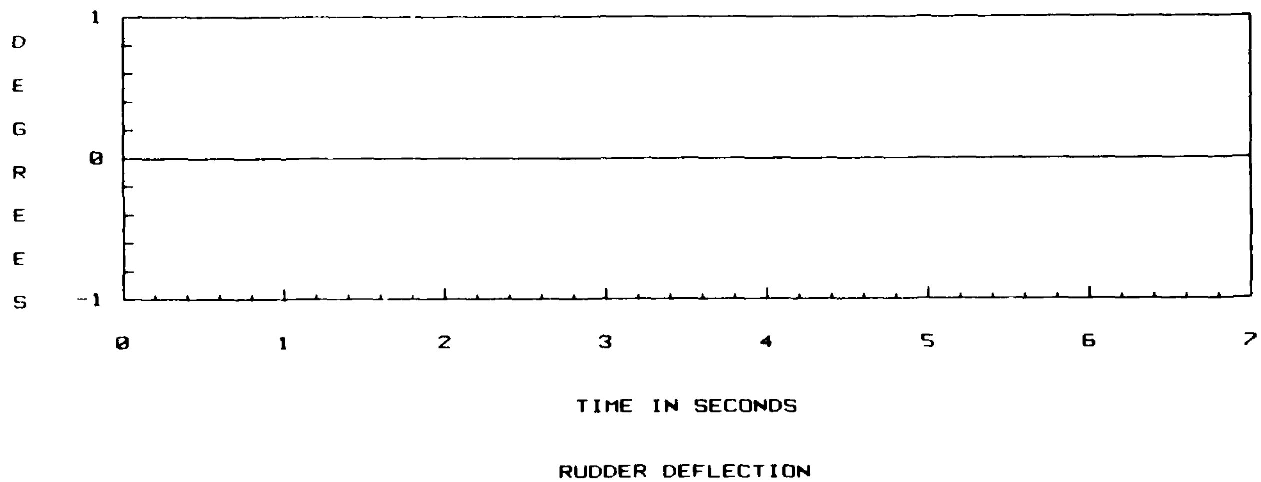


Figure E.31. Trailing Edge Deflection and Rates - θ_{cmd} (Discrete using Step-Response Matrix)



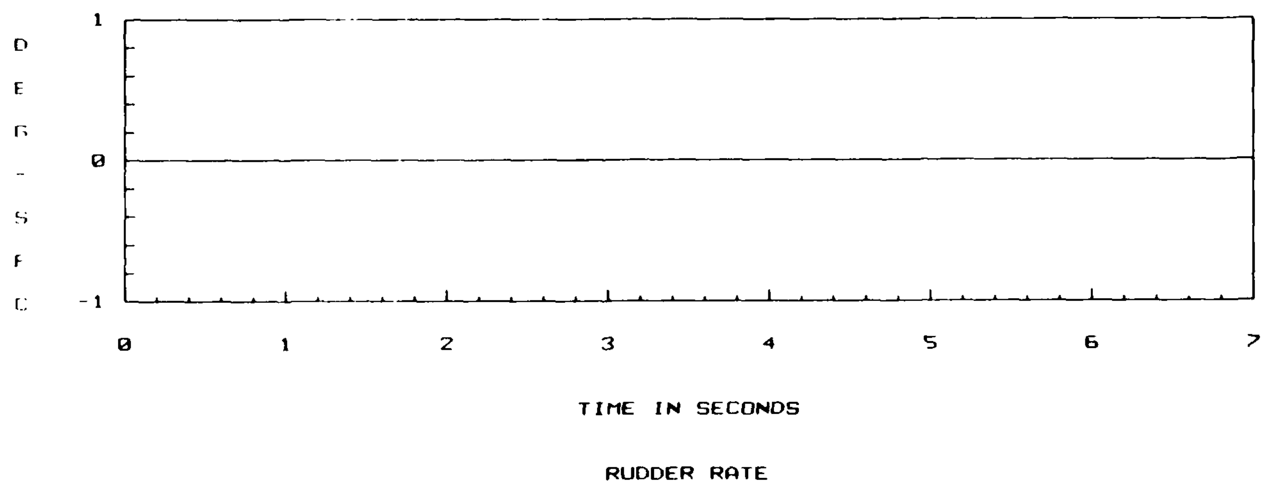
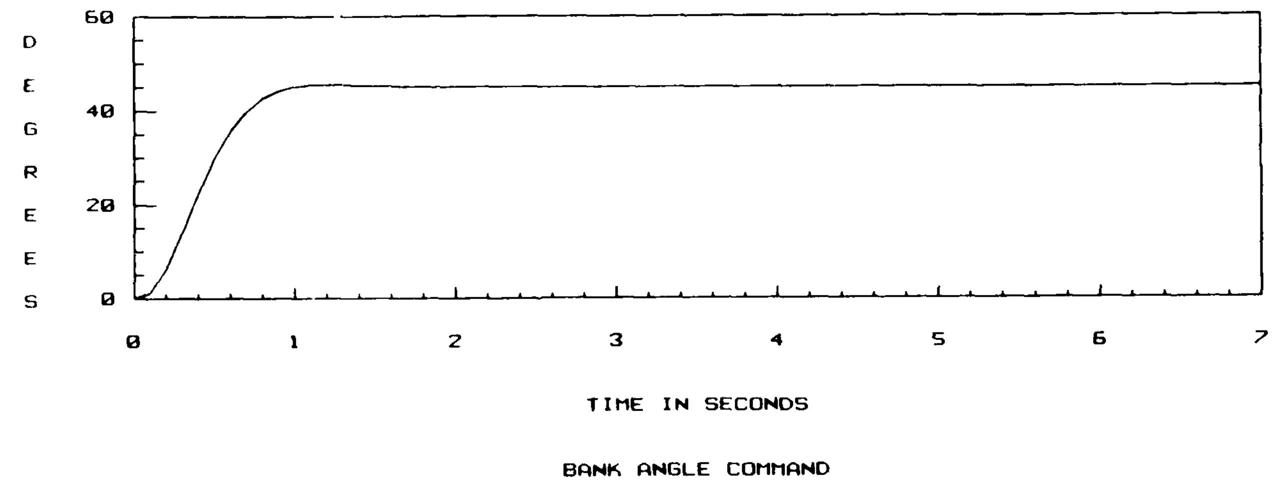


Figure E.32. Rudder Deflection and Rate - θ_{cmd} (Discrete using Step-Response Matrix)

Model Following Input Commands

Responses for the 45 degree coordinated turn maneuver are show for the model following input commands starting with Figure E.34 and illustrate the performance of the controller.



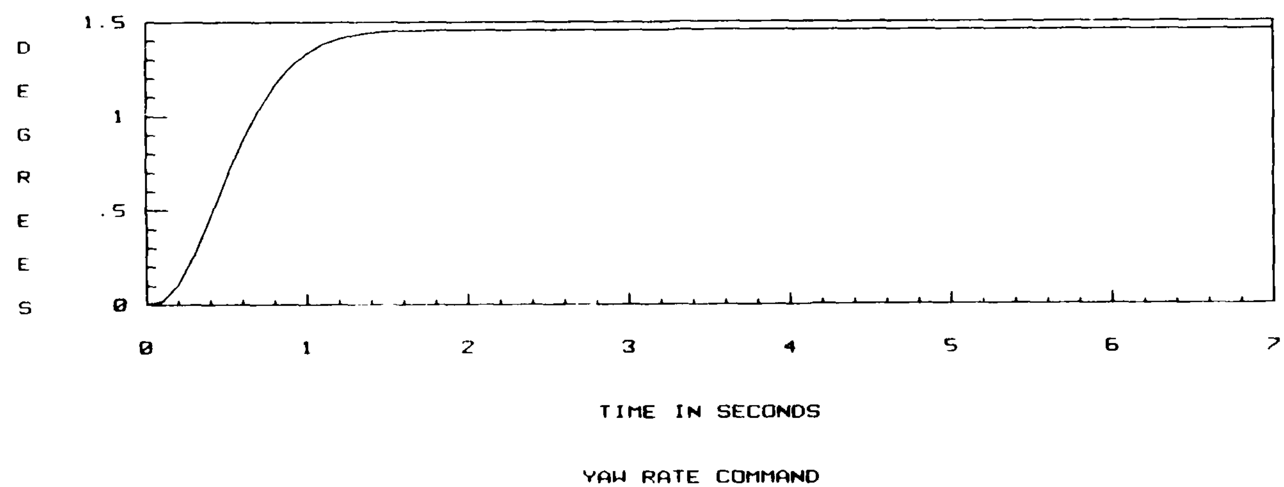


Figure E.33. ϕ_{cmd} and r_{cmd} - Model Following

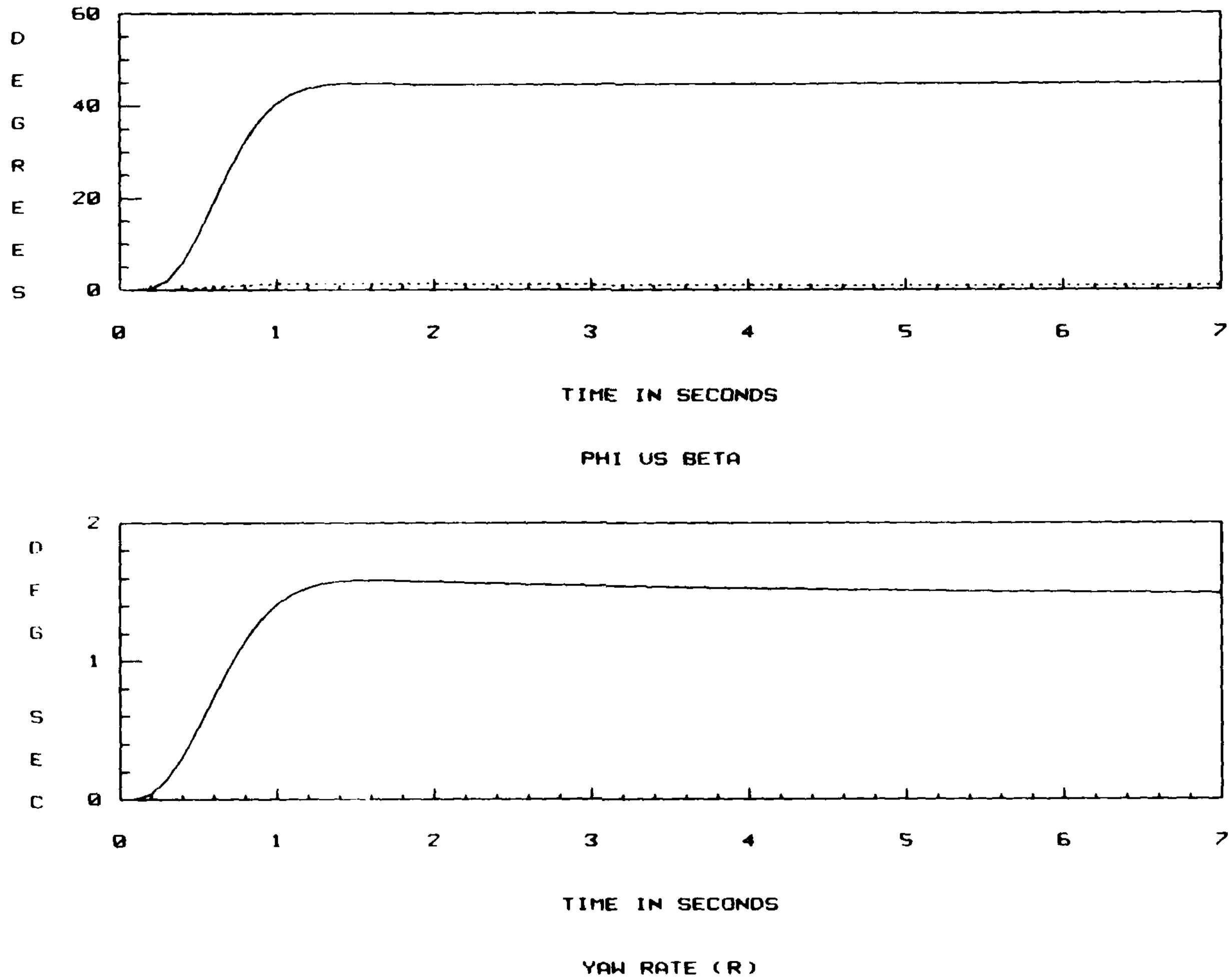


Figure E.34. ϕ, β, and r - 45° Banked Turn - Model Following (Continuous)

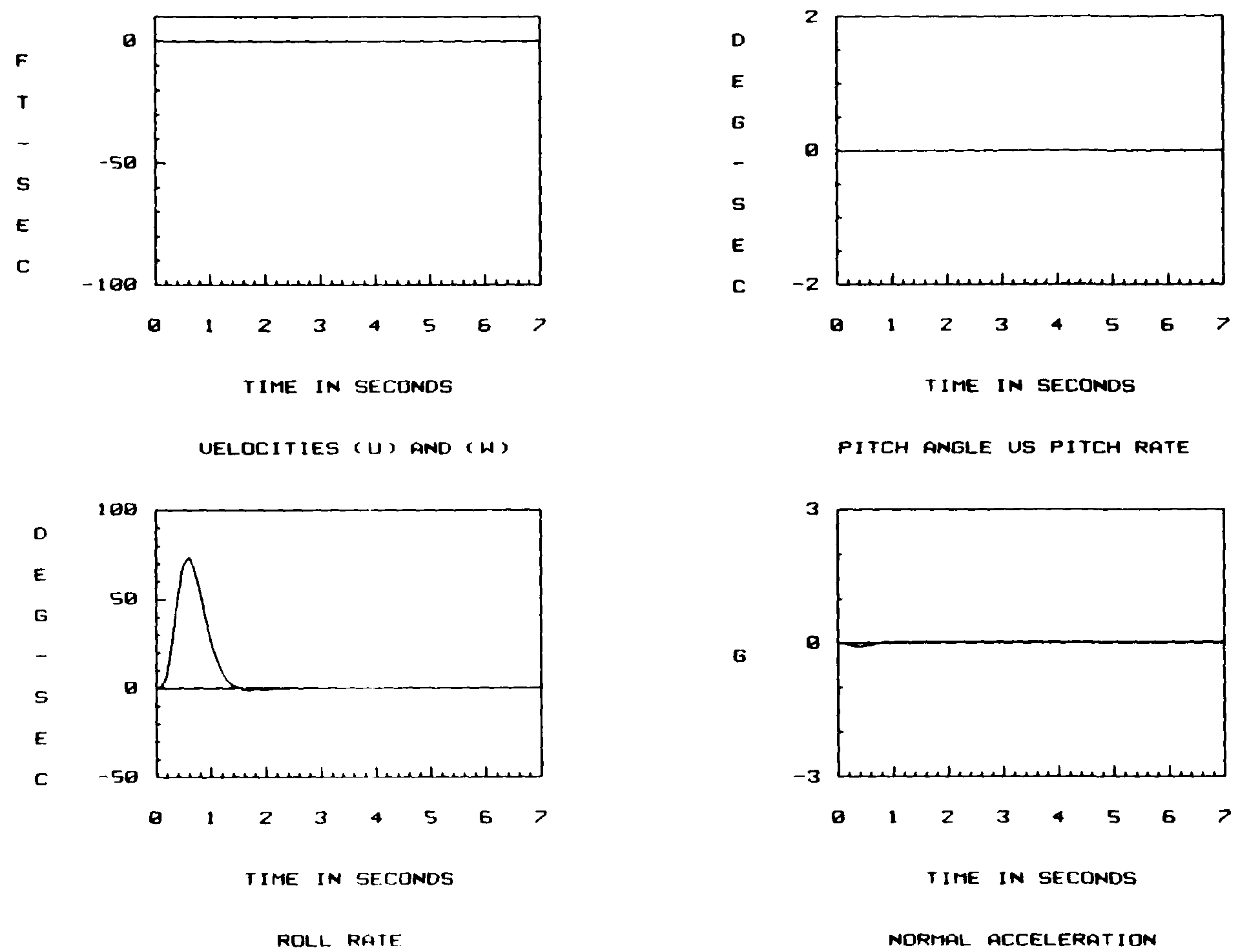


Figure E.35. u, w, θ, q, p, and Nz - 45° Banked Turn - Model Following (Continuous)

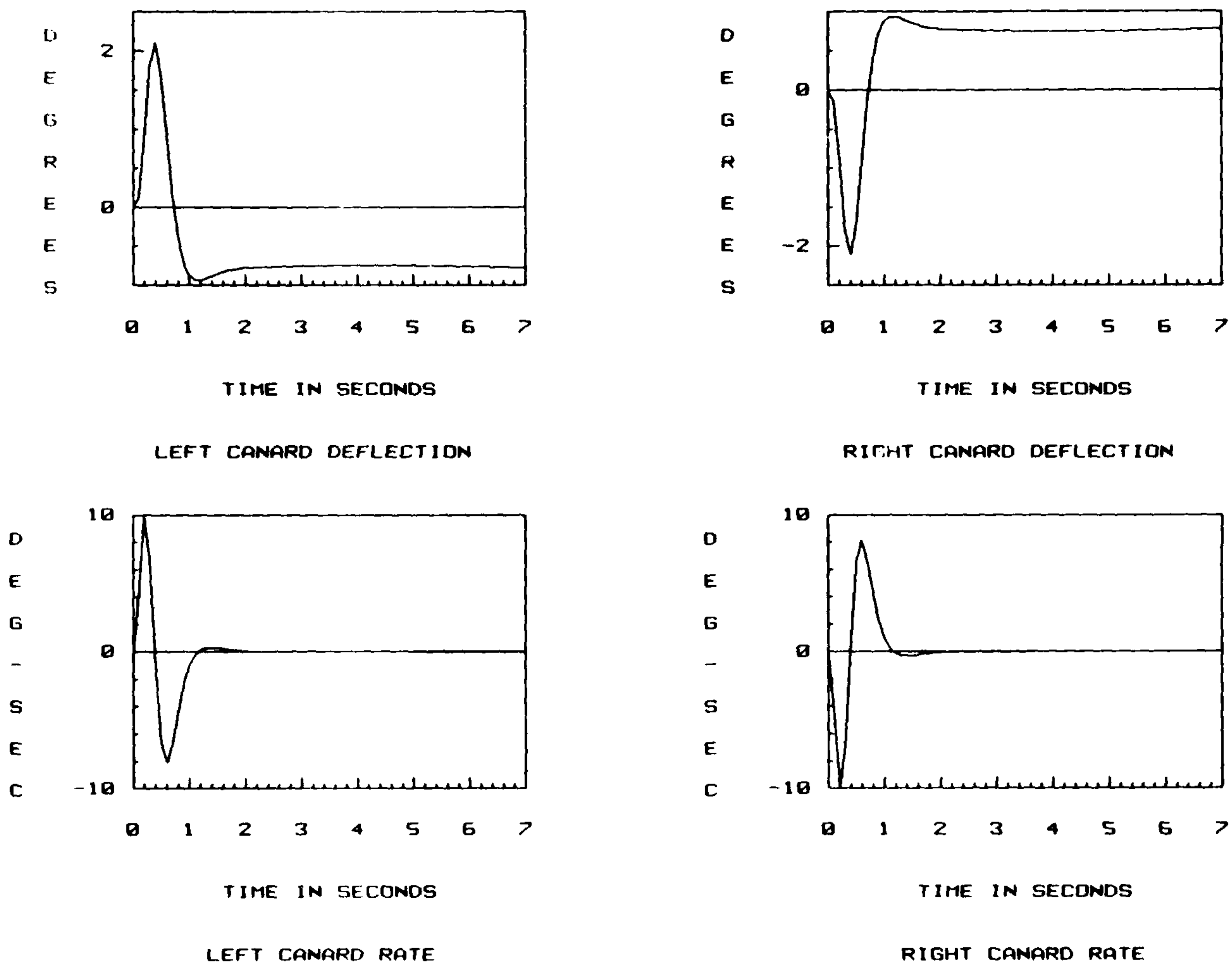


Figure E.36. Canard Deflection and Rates - 45° Banked Turn - Model Following (Continuous)

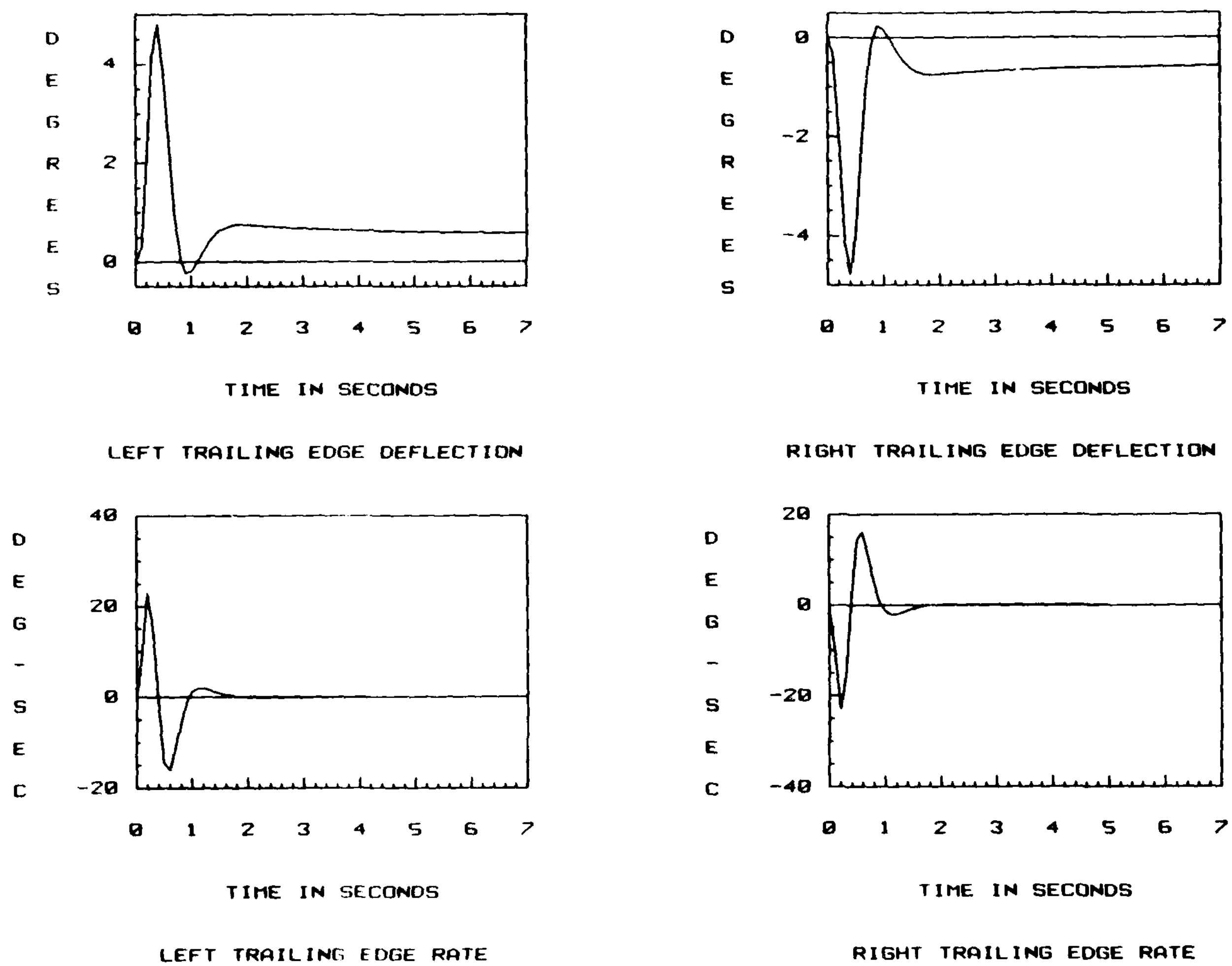


Figure E.37. Trailing Edge Deflection and Rates- 45° Banked Turn - Model Following (Continuous)

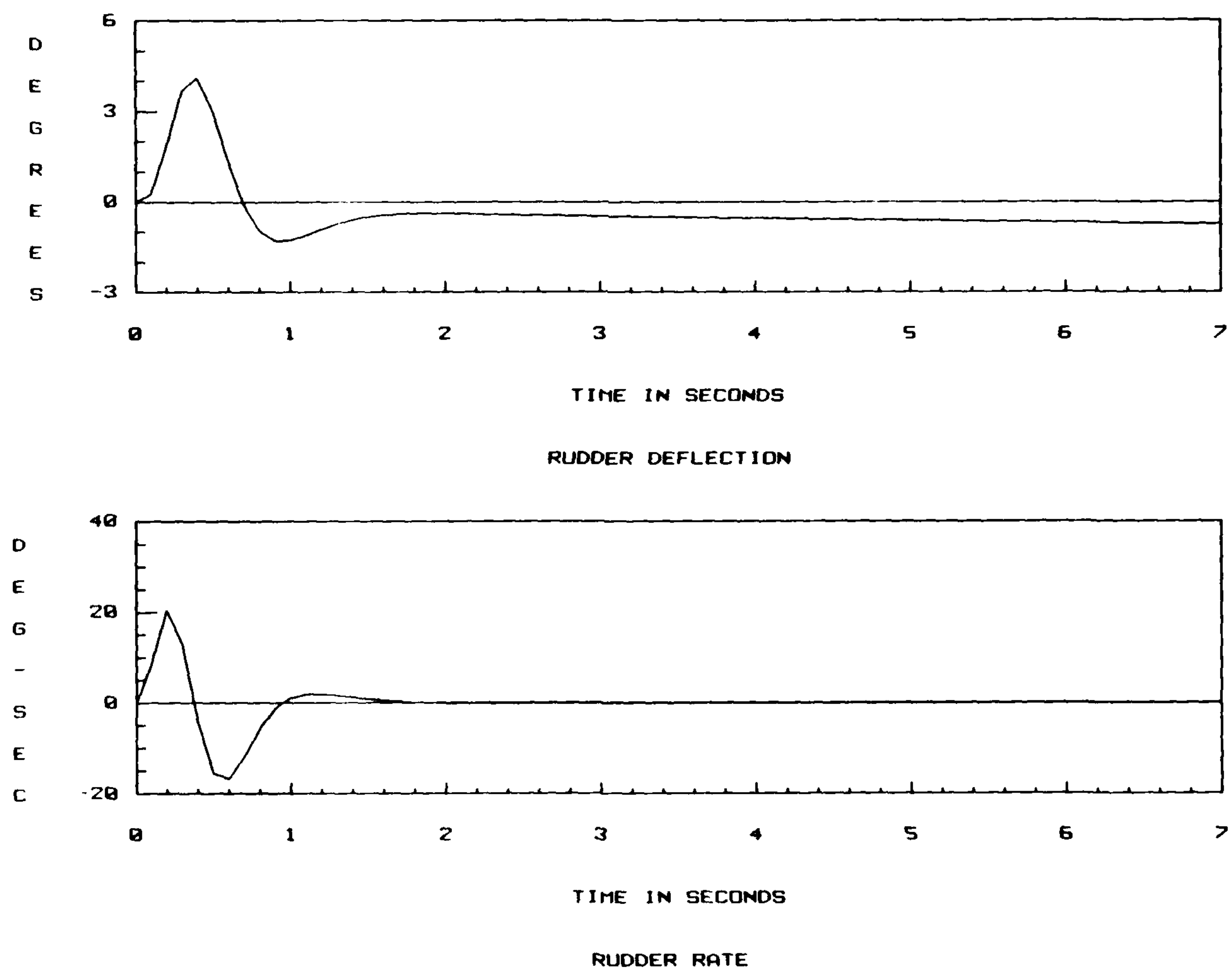


Figure E.38. Rudder Deflection and Rate - 45° Banked Turn - Model Following (Continuous)

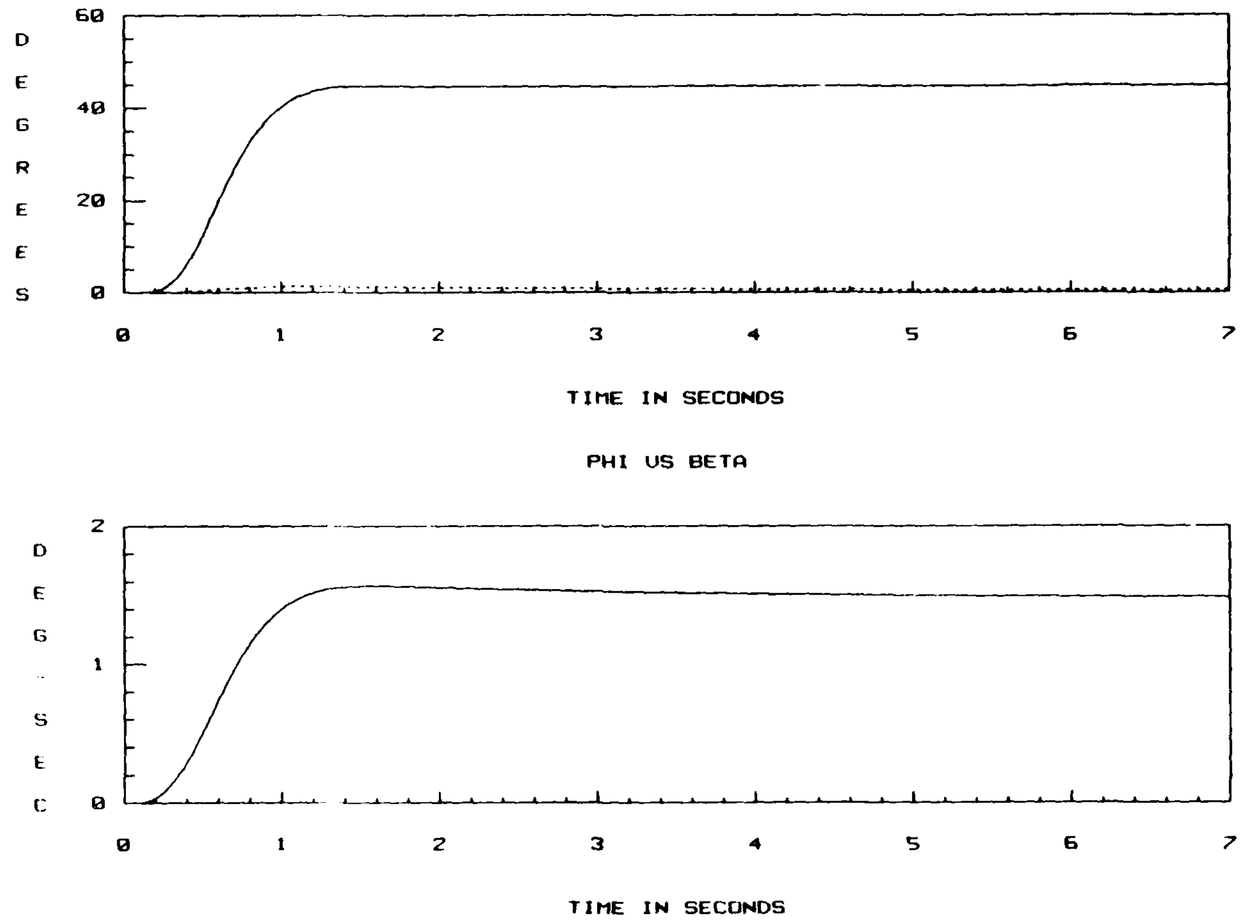


Figure E.39. ϕ, β, and r - 45° Banked Turn - Model Following (Discrete)

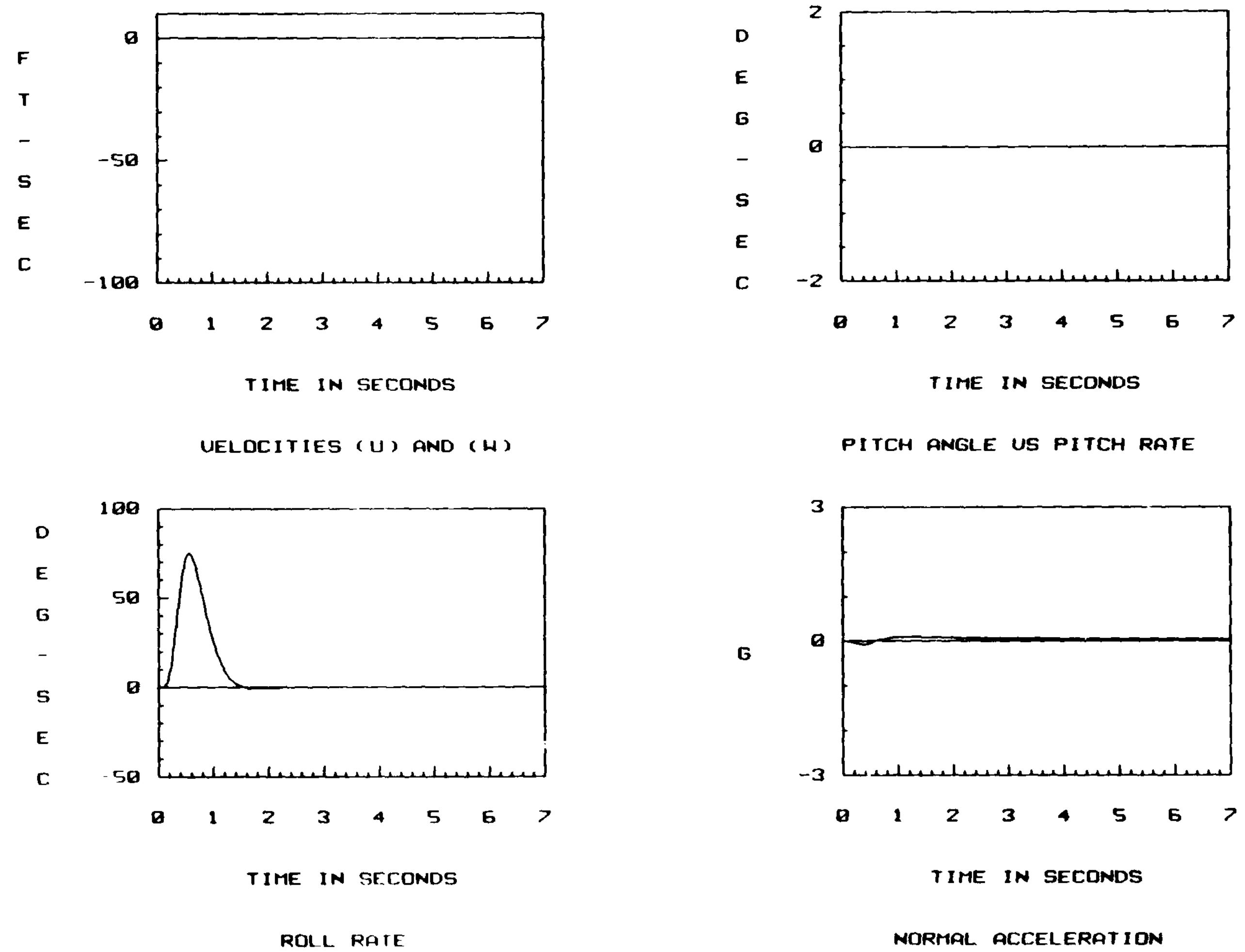


Figure E.40. u, w, θ, q, p, and Nz - 45° Banked Turn - Model Following (Discrete)

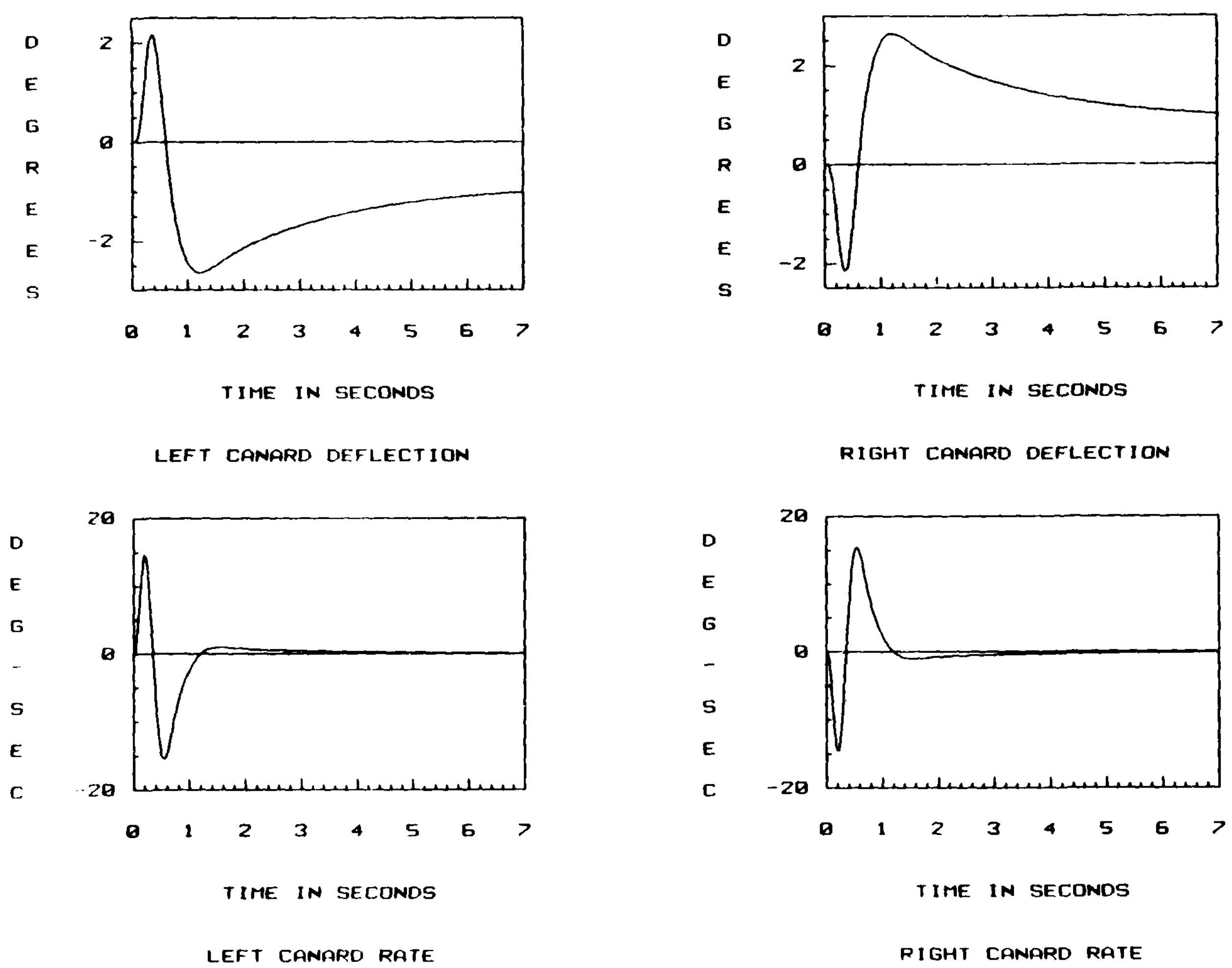


Figure E.41. Canard Deflection and Rates - 45° Banked Turn - Model Following (Discrete)

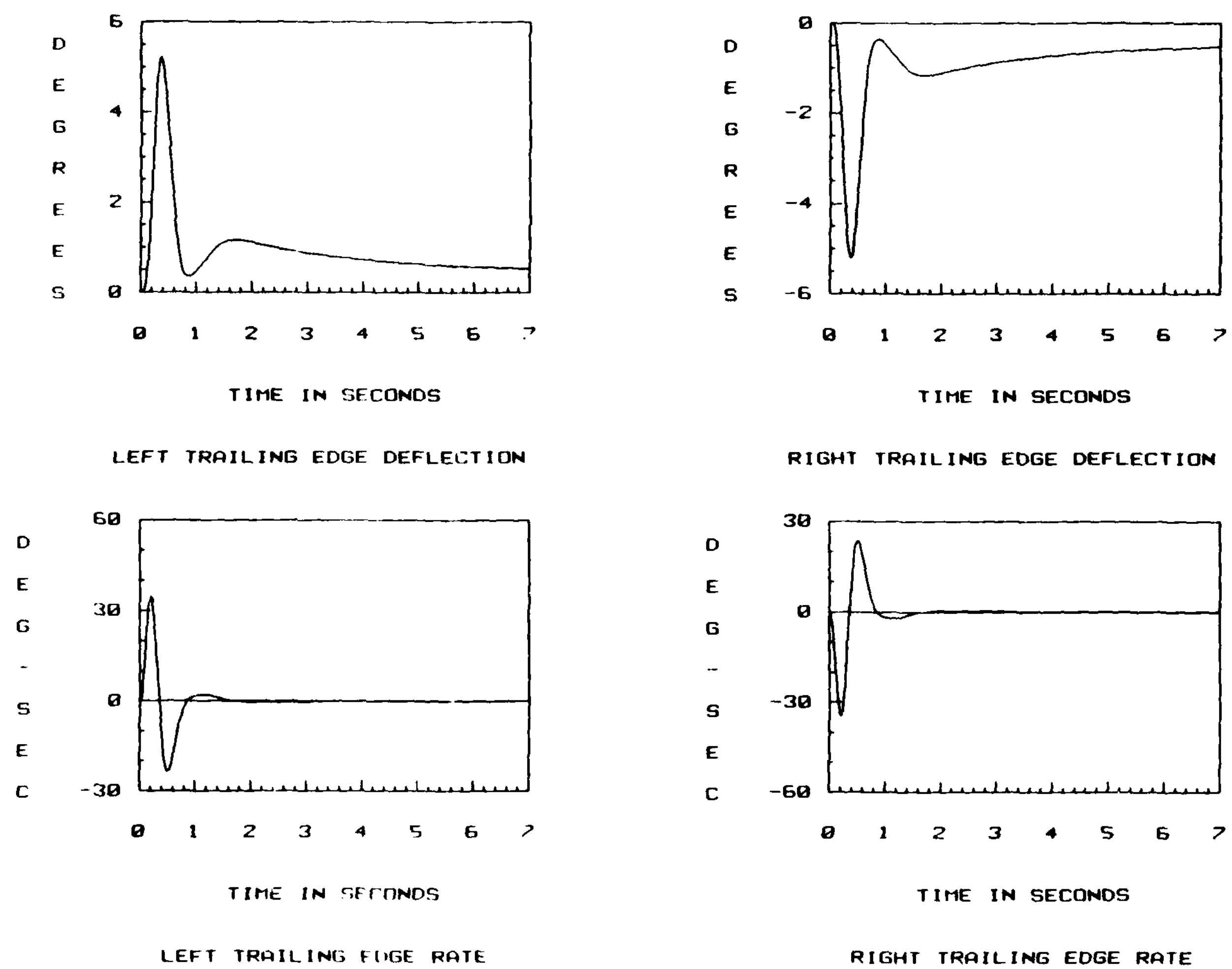


Figure E.42. Trailing Edge Deflection and Rates- 45° Banked Turn - Model Following (Discrete)

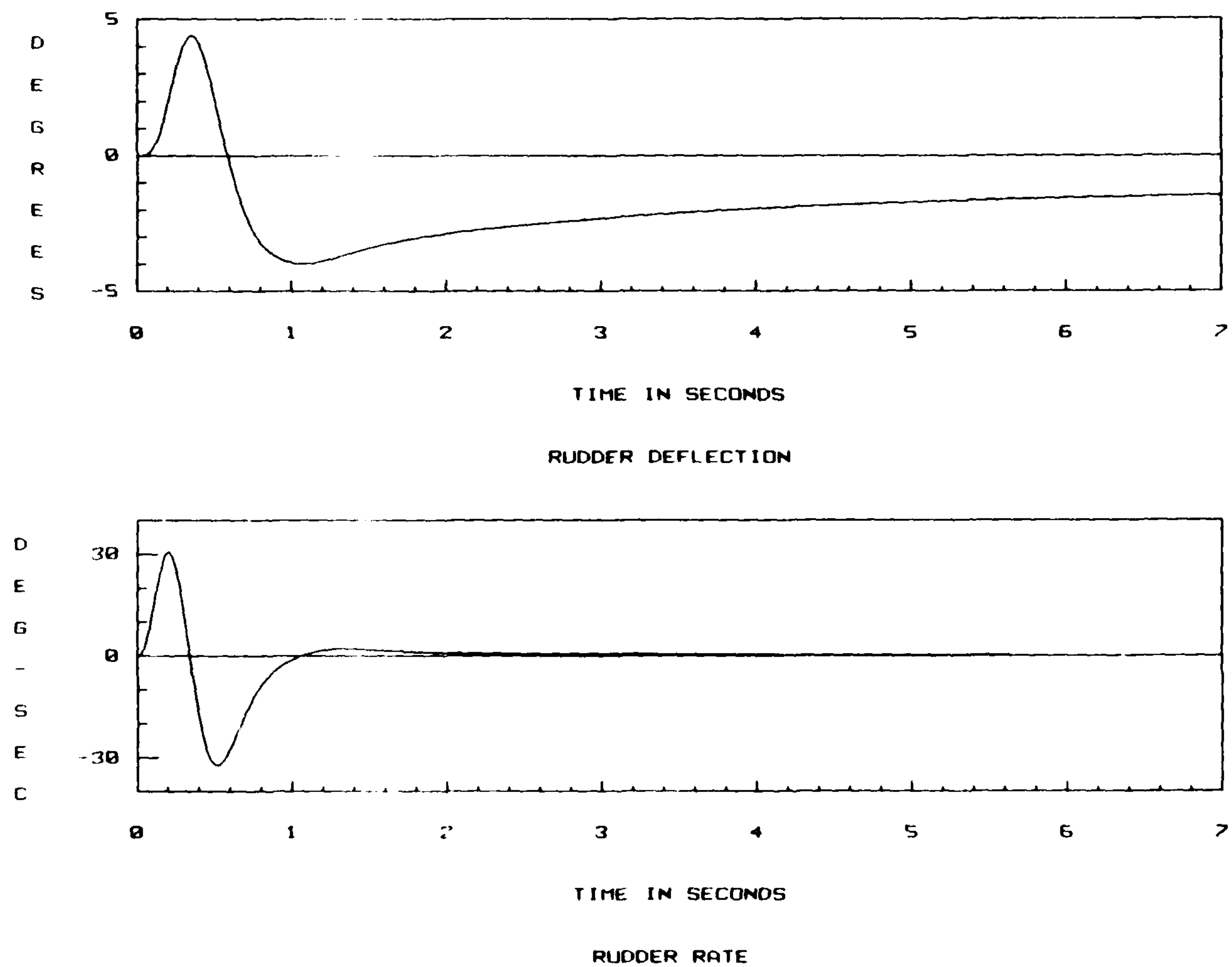
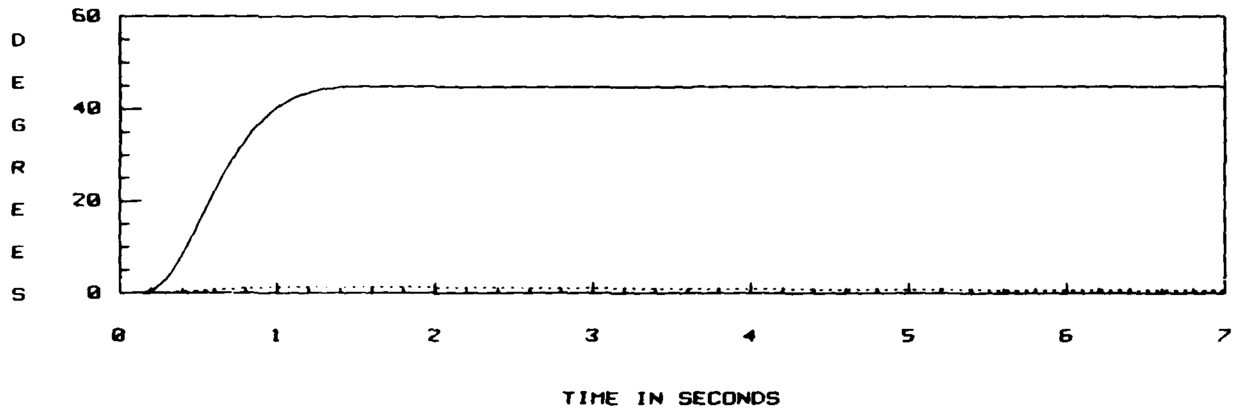
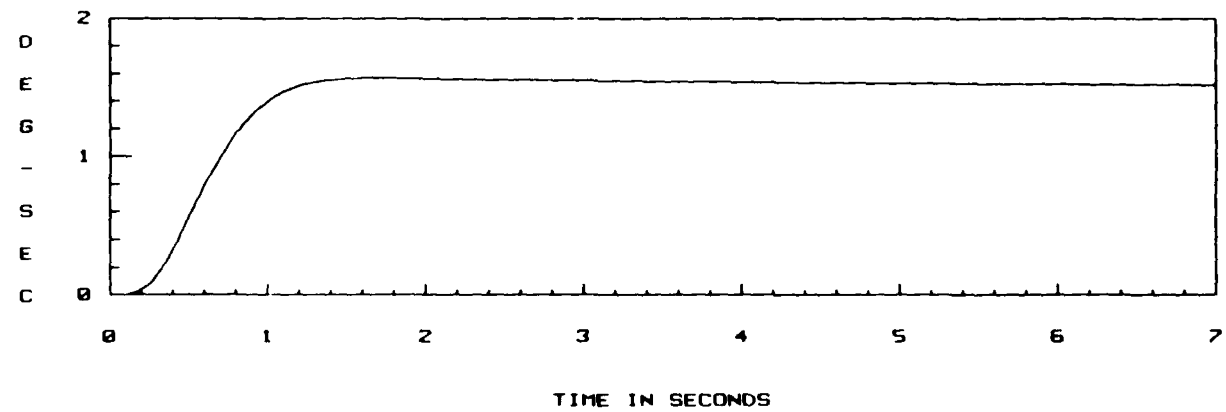


Figure E.43. Rudder Deflection and Rate - 45° Banked Turn - Model Following (Discrete)



PHI VS BETA



YAW RATE (R)

Figure E.44. ϕ, β, and r - 45° Banked Turn - Model Following (Discrete using Step-Response Matrix)

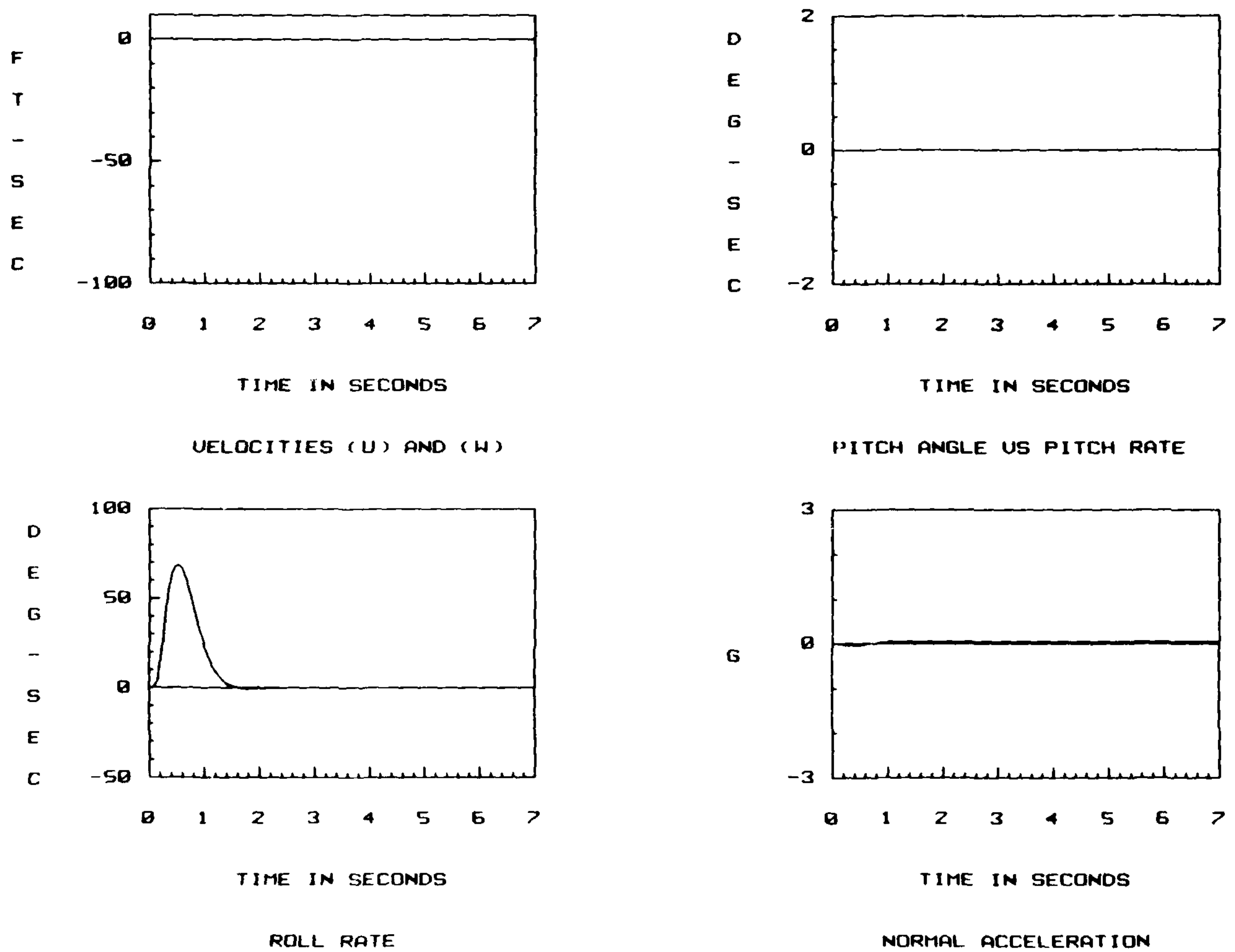


Figure E.45. u, w, θ, q, p, and Nz - 45° Banked Turn - Model Following (Discrete using Step-Response Matrix)

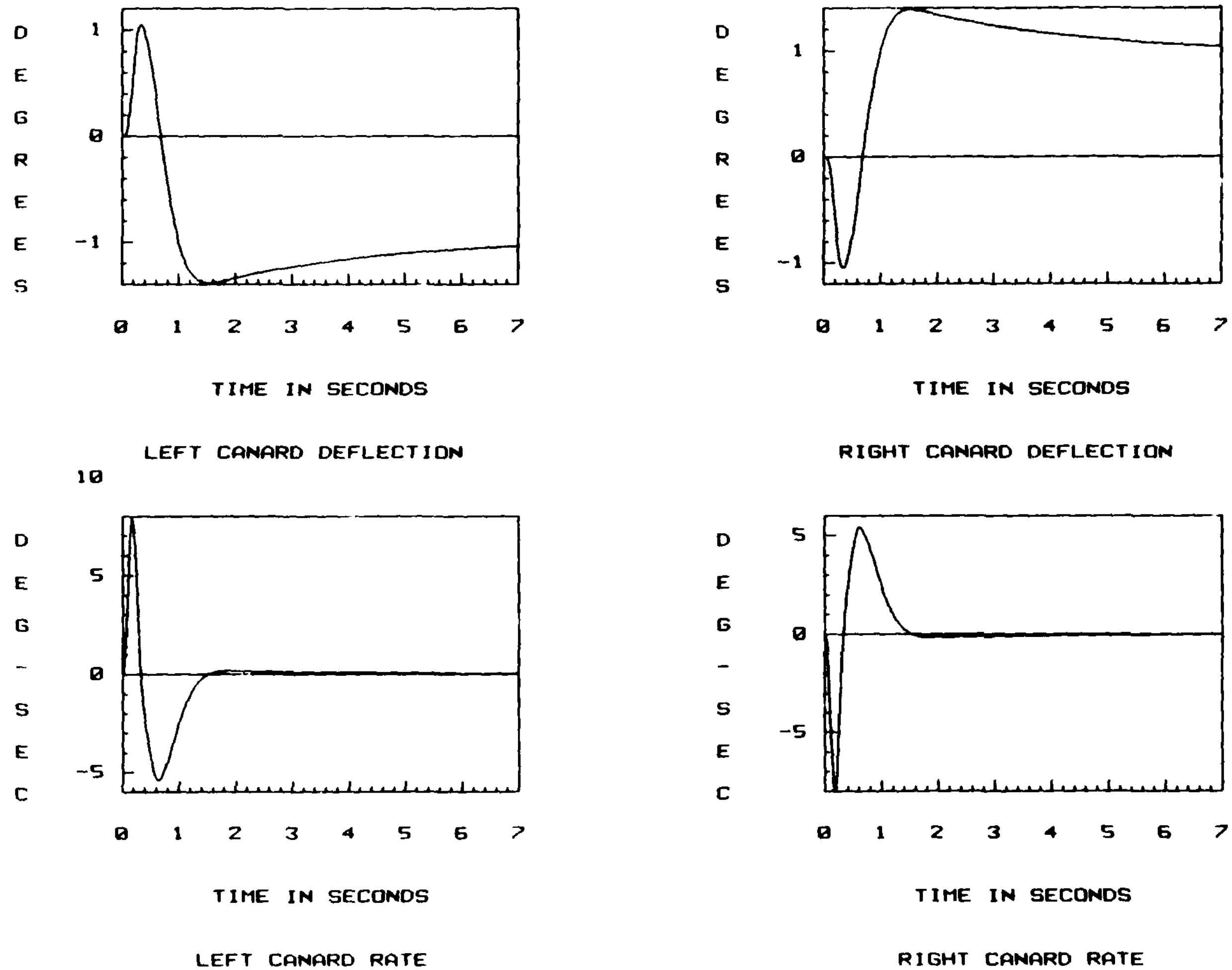


Figure E.46. Canard Deflection and Rates - 45° Banked Turn - Model Following (Discrete using Step-Response Matrix)

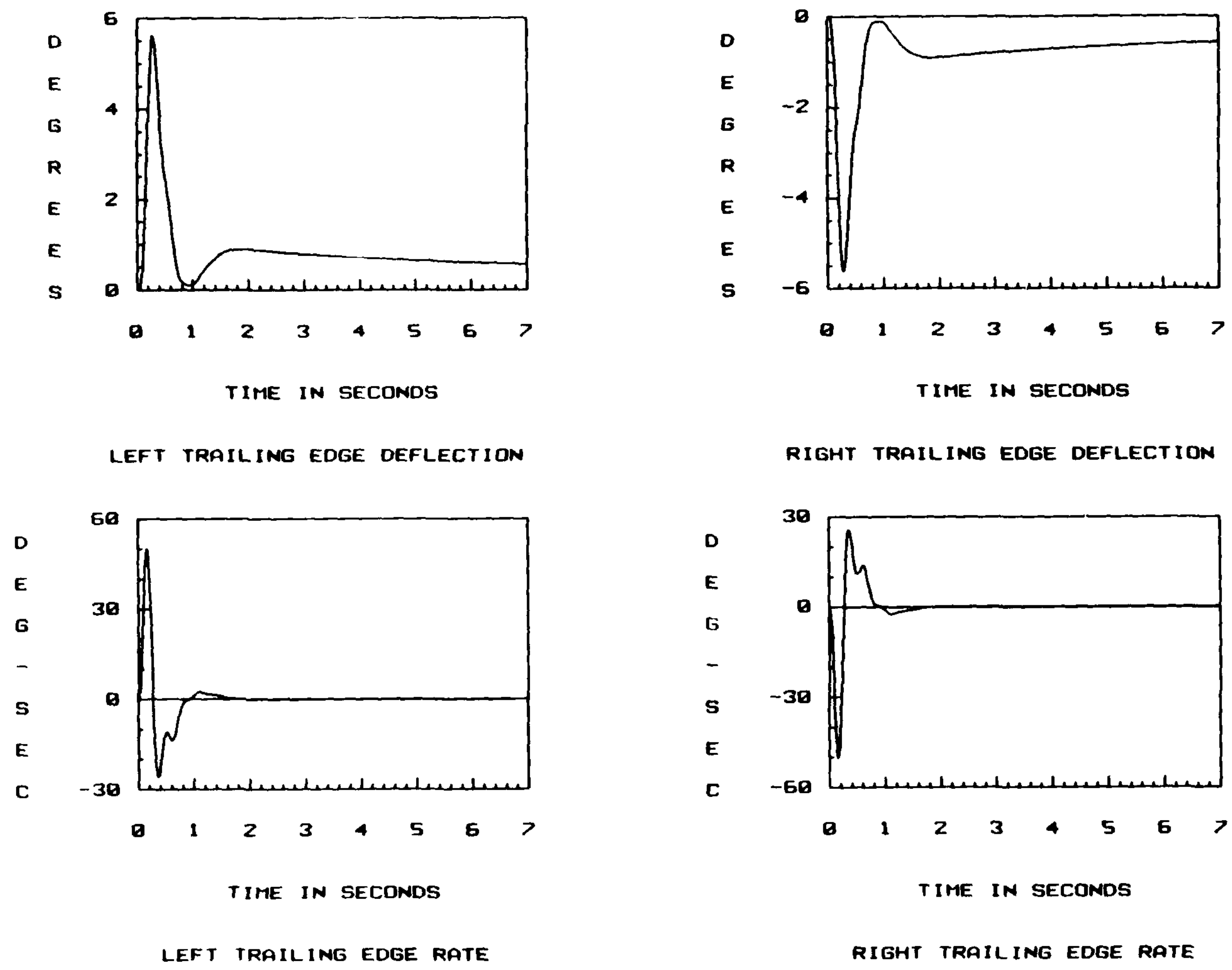
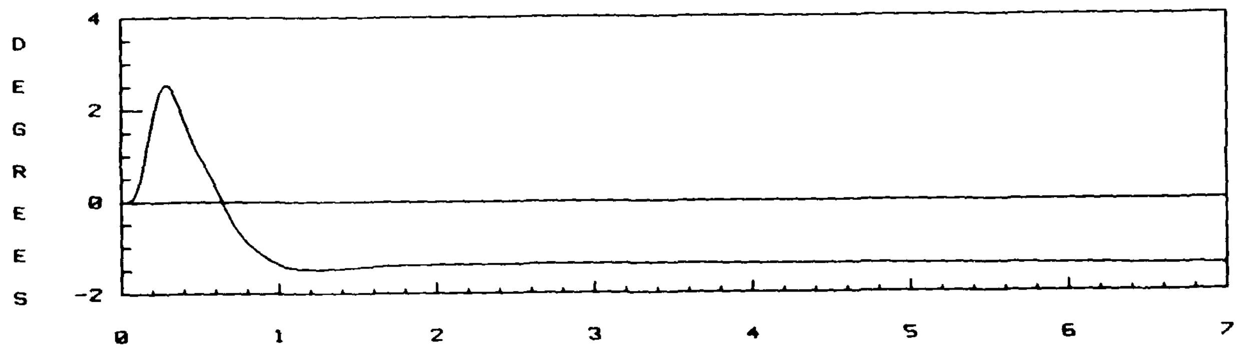
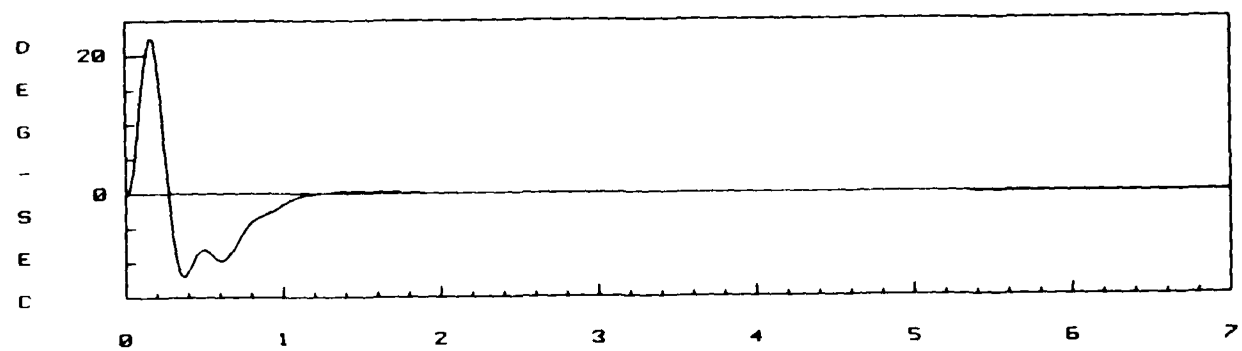


Figure E.47. Trailing Edge Deflection and Rates- 45° Banked Turn - Model Following (Discrete using Step-Response Matrix)



RUDDER DEFLECTION



RUDDER RATE

Figure E.48. Rudder Deflection and Rate - 45° Banked Turn - Model Following (Discrete using Step-Response Matrix)

Responses for the pitch tracking command are show starting with Figure E.49 and illustrate the performance of the controller.

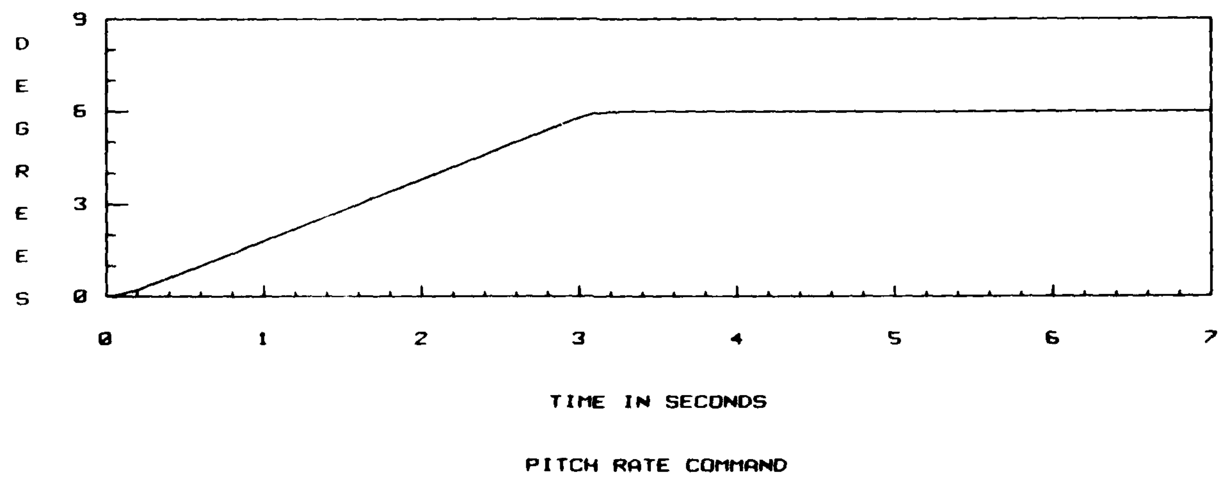


Figure E.49. θ_{cmd} - Model Following

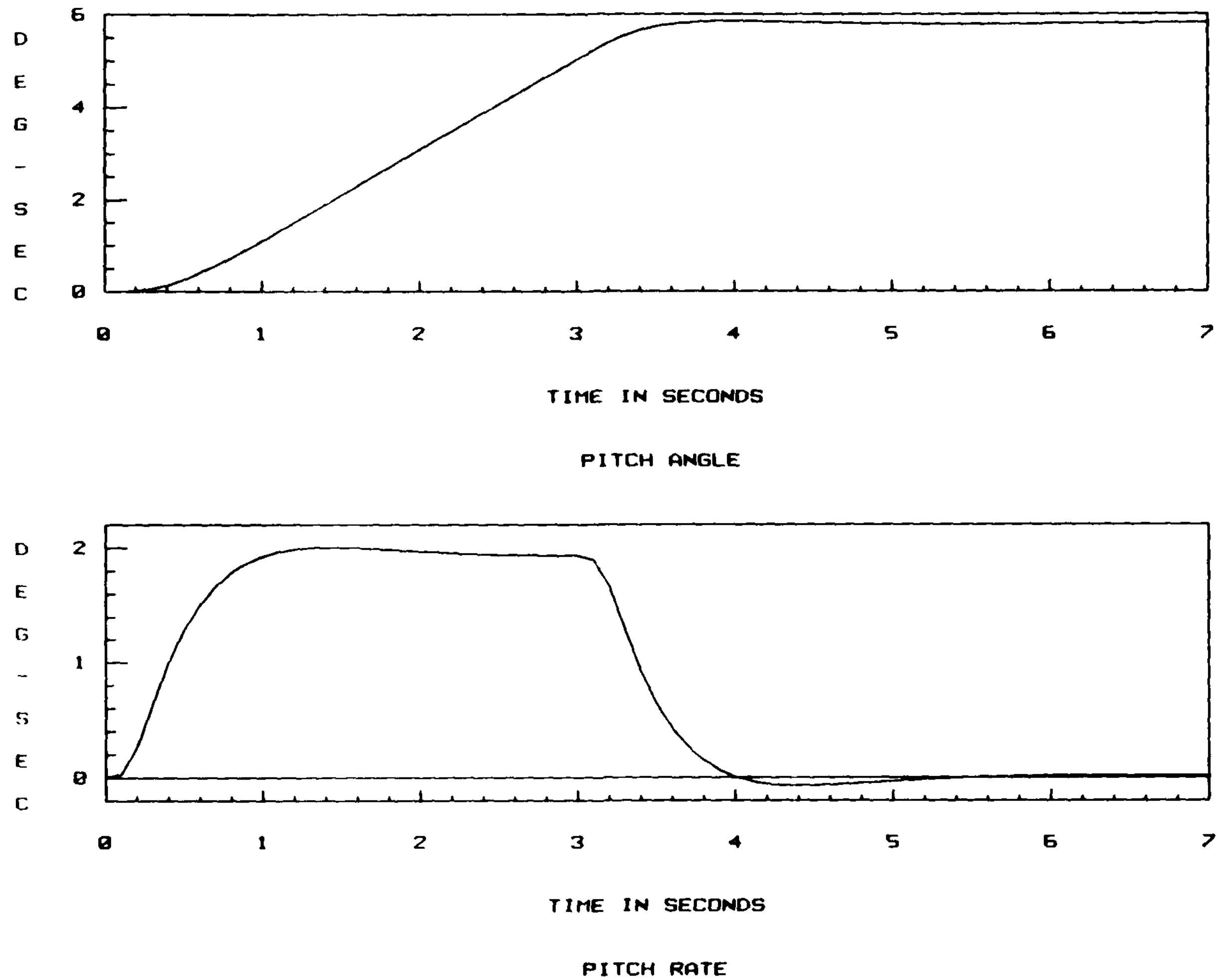


Figure E.50. θ and q - θ_{cmd} - Model Following (Continuous)

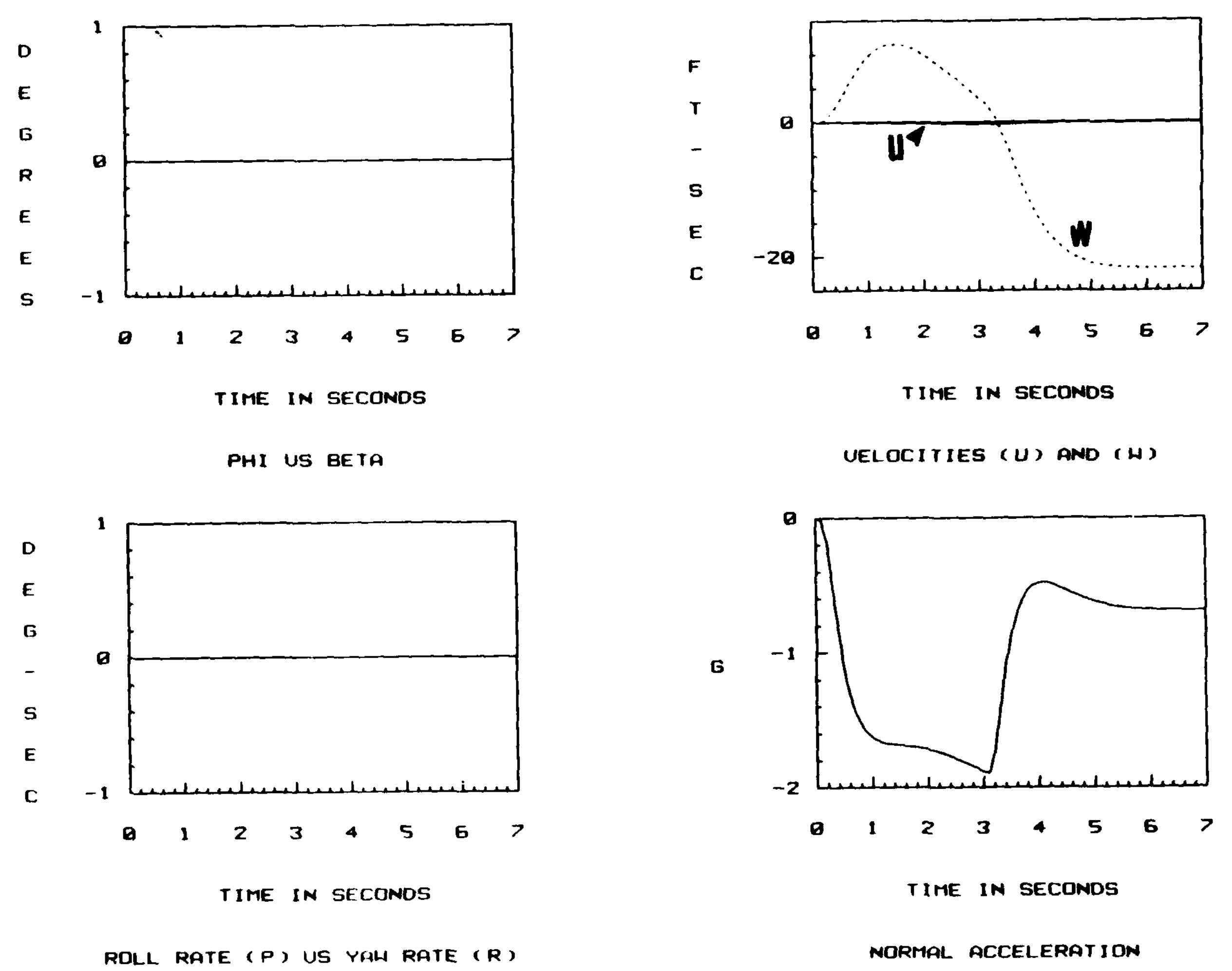


Figure E.51. ϕ, β, u, w, r, p, and Nz - θ_{cmd} - Model Following (Continuous)

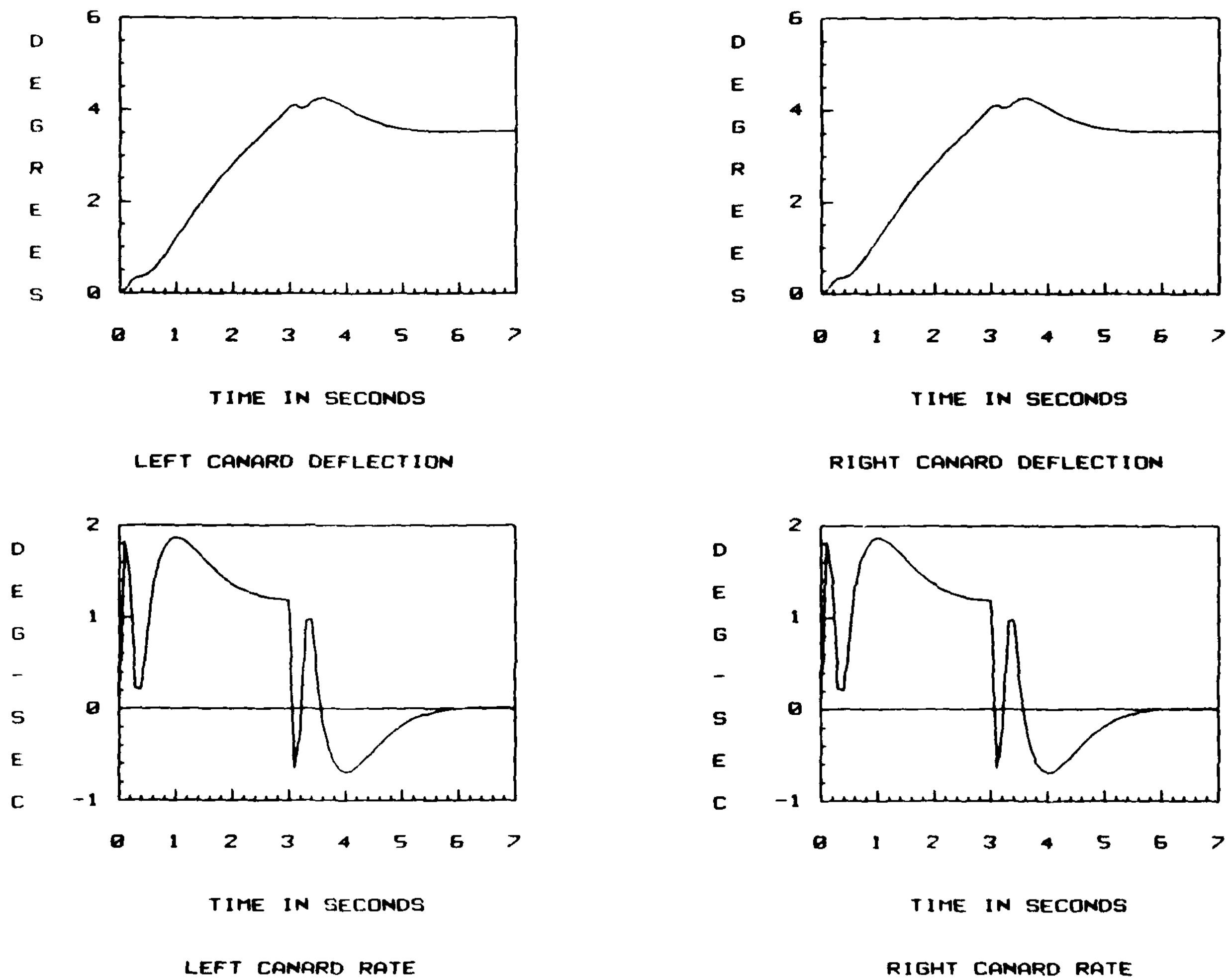


Figure E.52. Canard Deflection and Rates - θ_{cmd} - Model Following (Continuous)

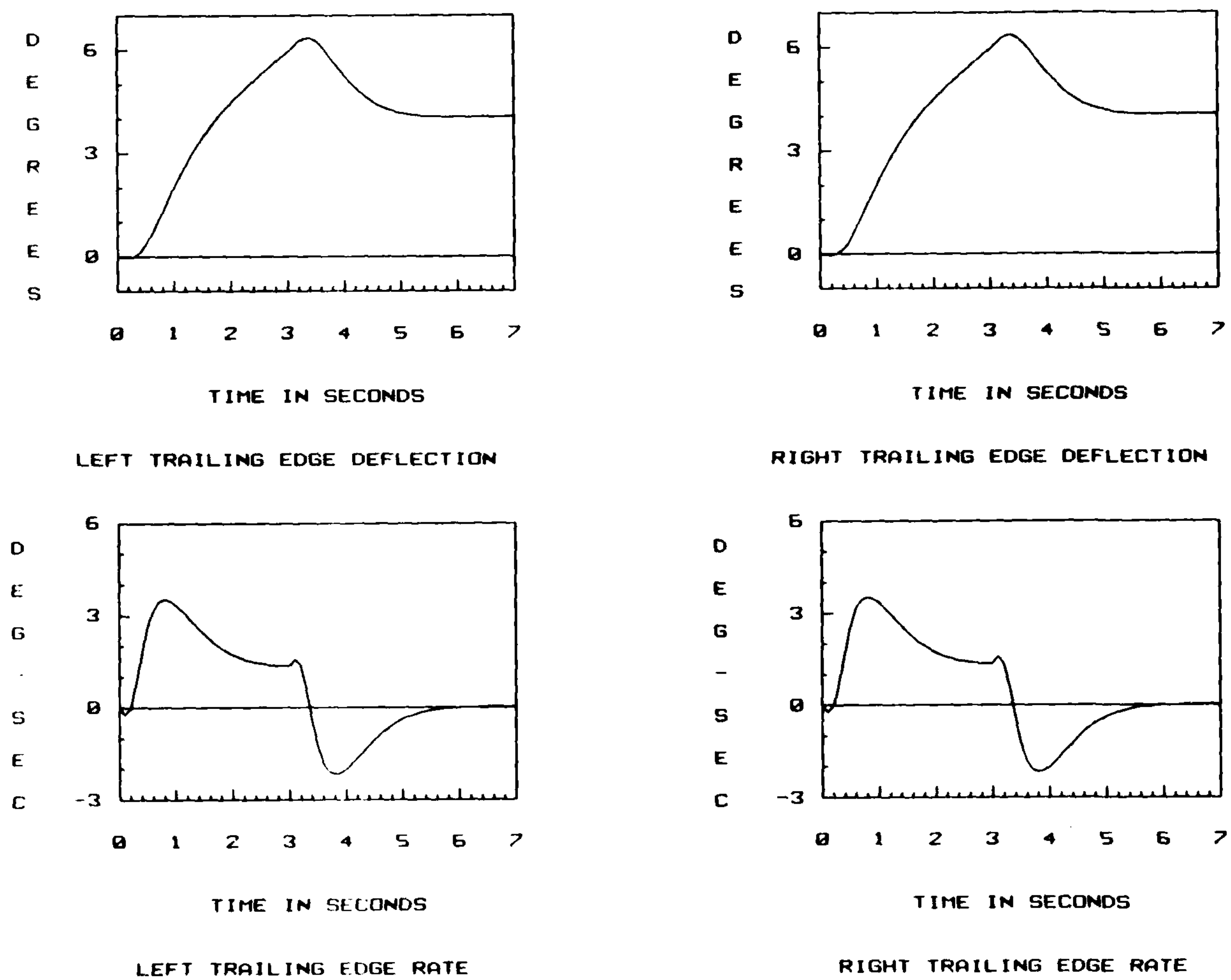
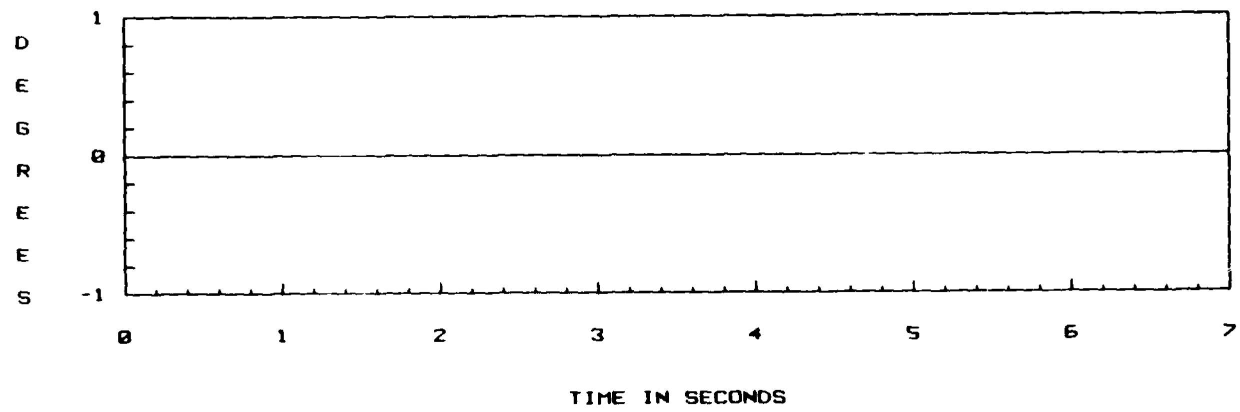


Figure E.53. Trailing Edge Deflection and Rates - θ_{cmd} - Model Following (Continuous)



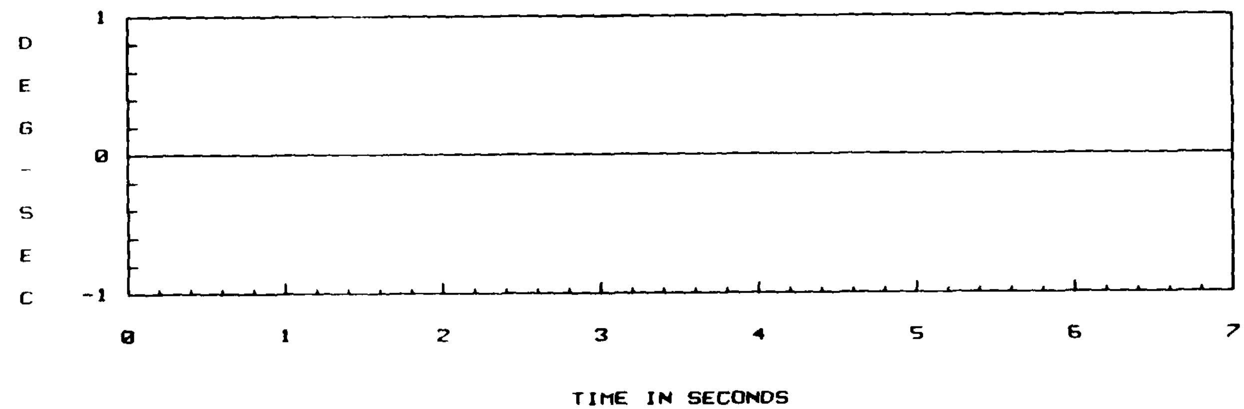


Figure E.54. Rudder Deflection and Rate - θ_{cmd} - Model Following (Continuous)

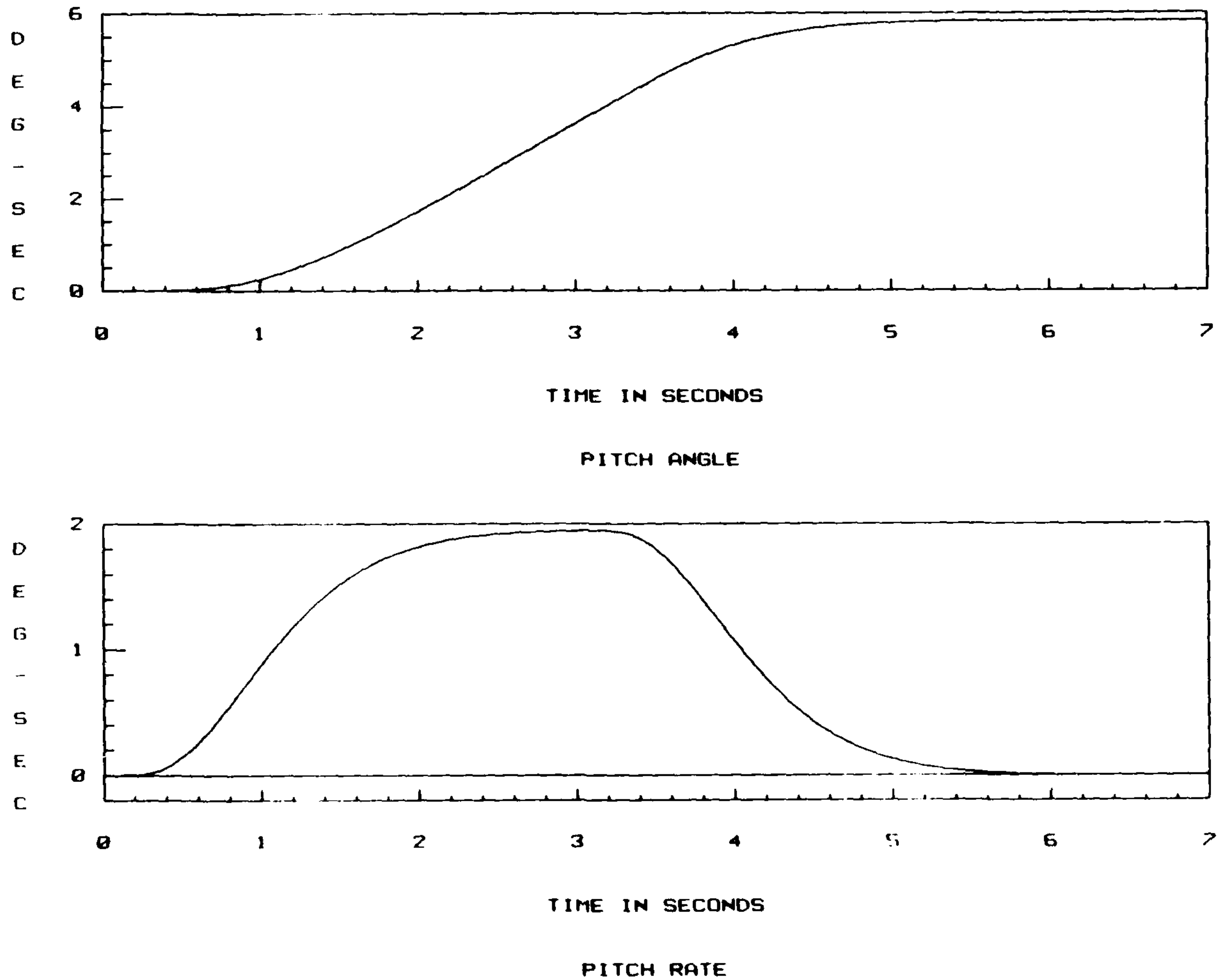


Figure E.55. θ and q - θ_{cmd} - Model Following (Discrete)

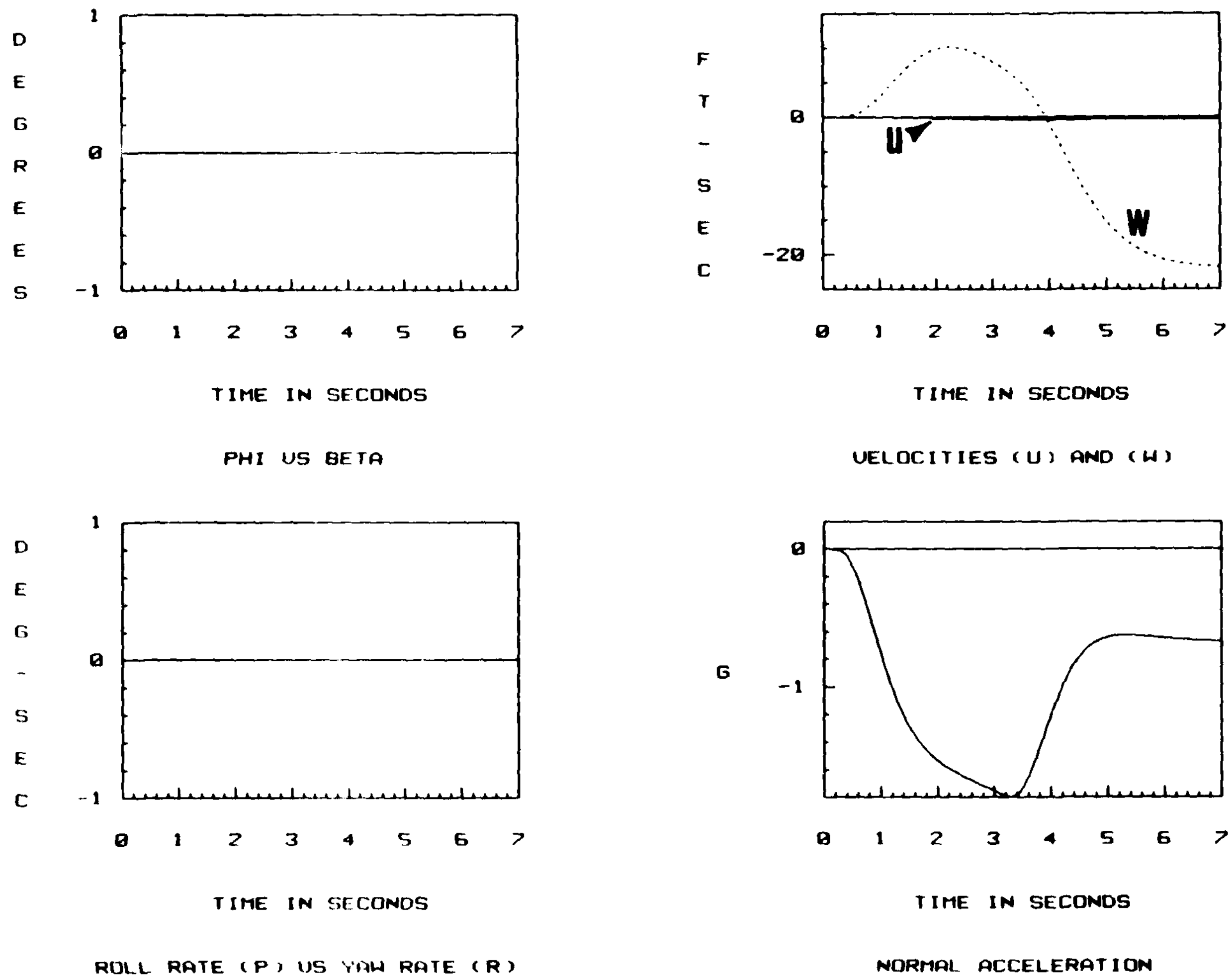


Figure E.56. ϕ, β, u, w, r, p, and Nz - θ_{cmd} - Model Following (Discrete)

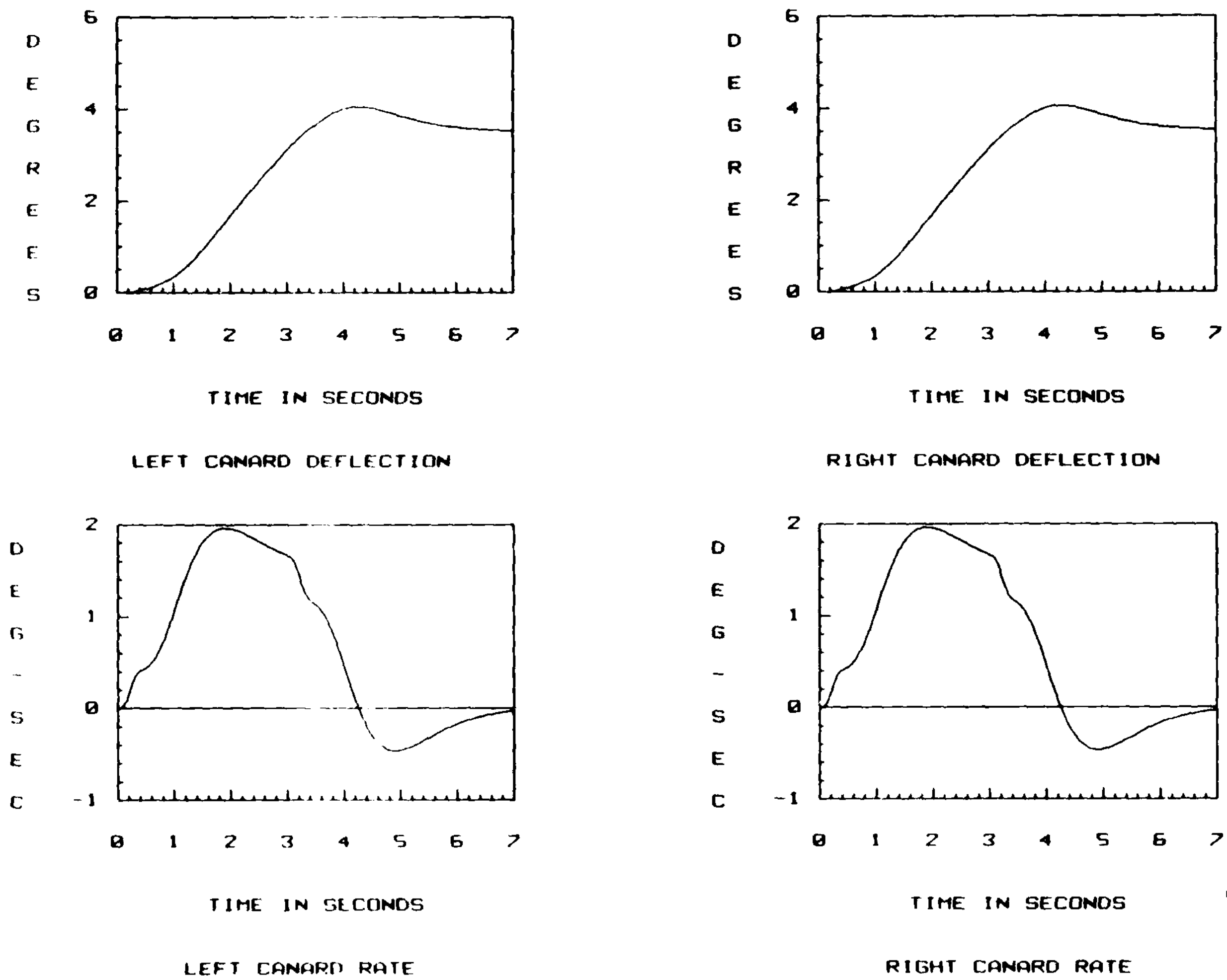


Figure E.57. Canard Deflection and Rates - θ_{cmd} - Model Following (Discrete)

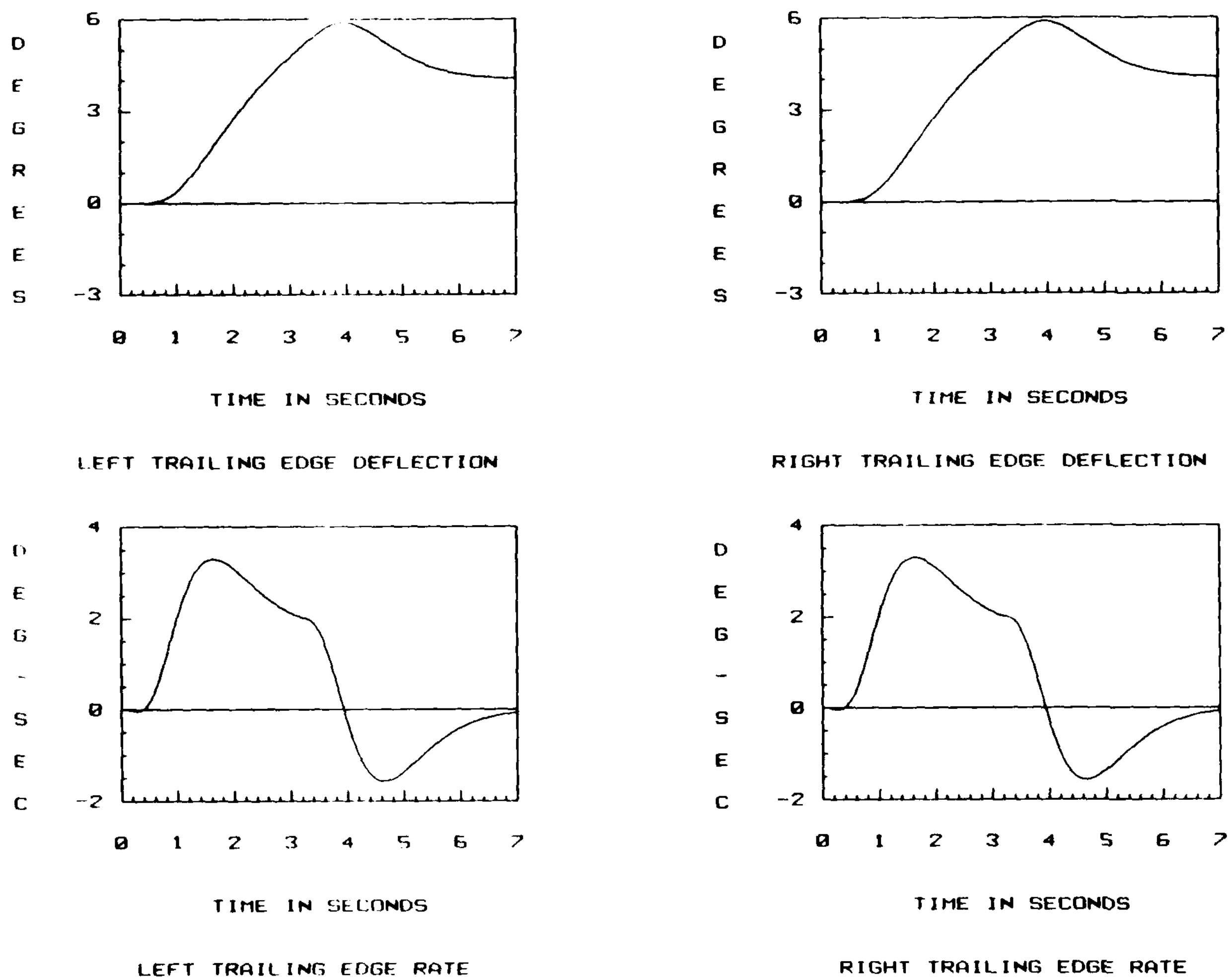


Figure E.58. Trailing Edge Deflection and Rates - θ_{cmd} - Model Following (Discrete)

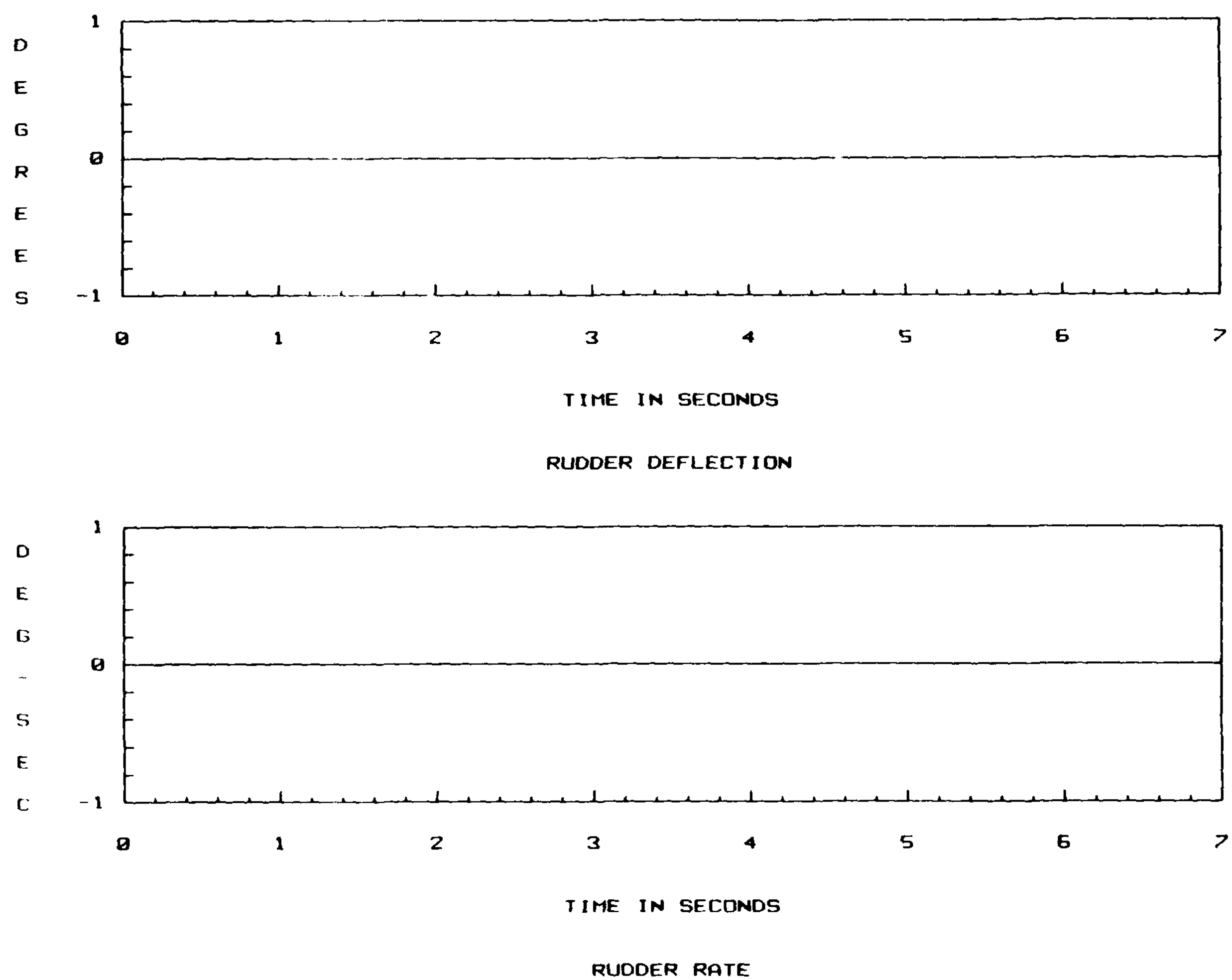


Figure E.59. Rudder Deflection and Rate - θ_{cmd} - Model Following (Discrete)

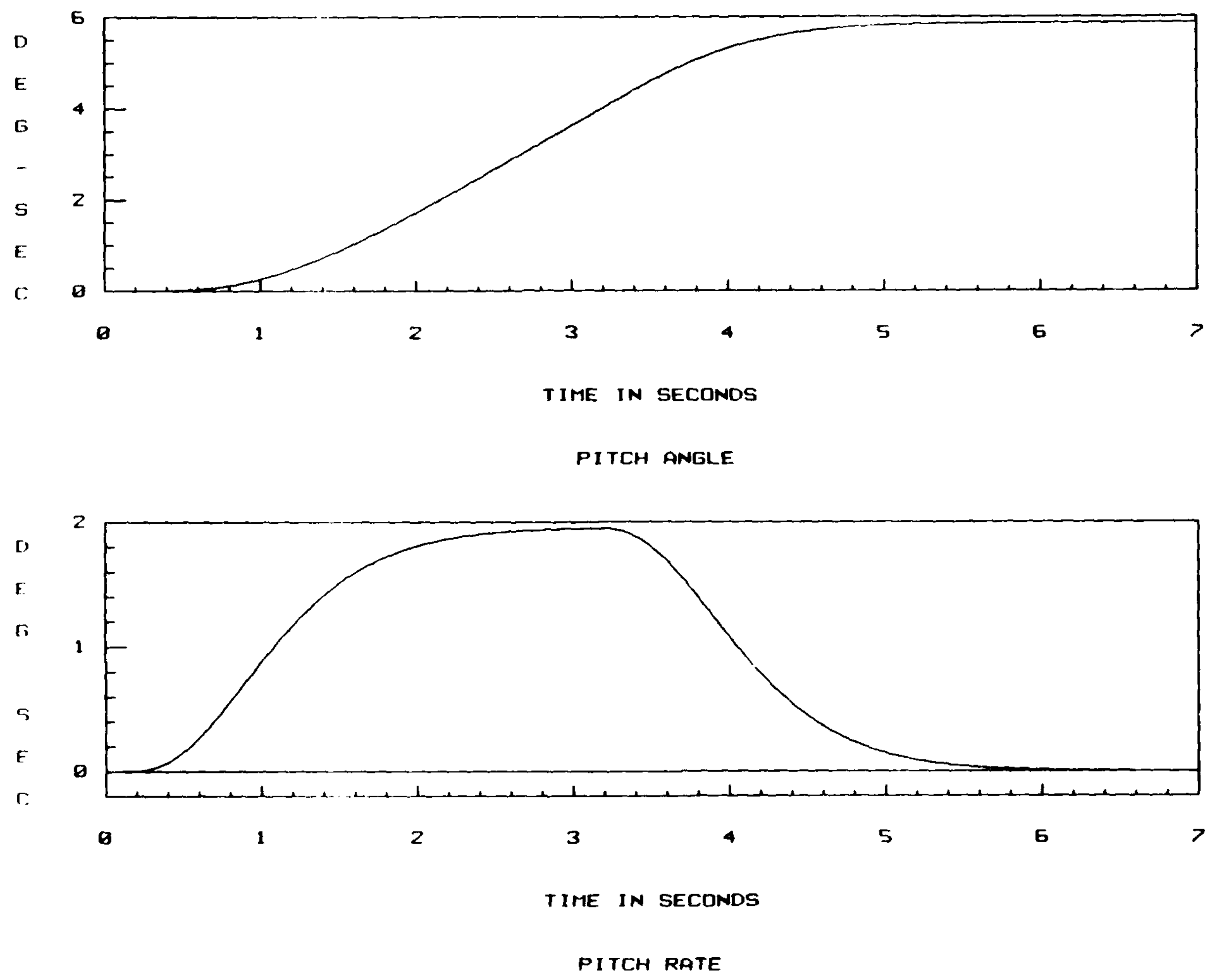


Figure E.60. θ and q - θ_{cmd} - Model Following (Discrete using Step-Response Matrix)

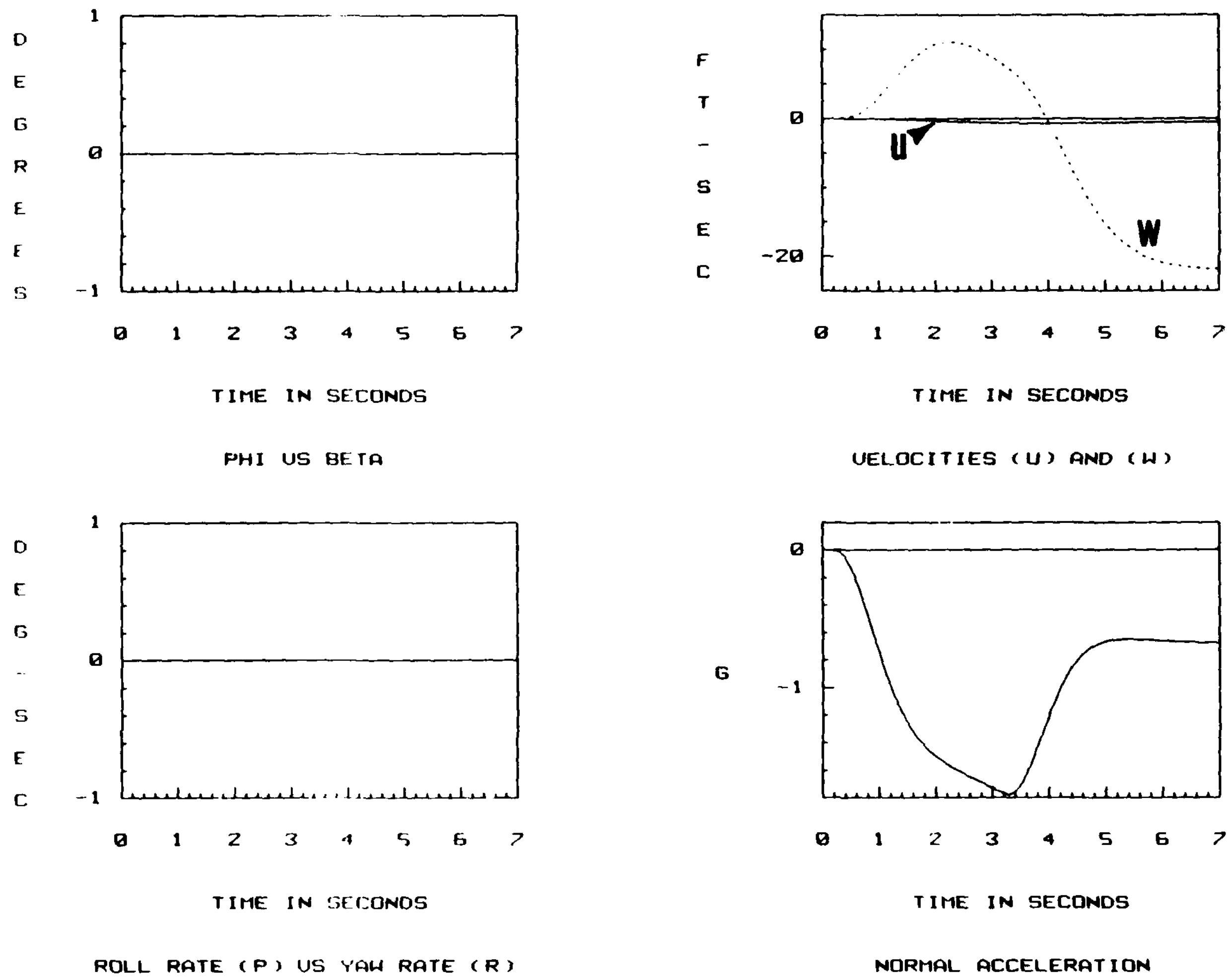


Figure E.61. ϕ, β, u, w, r, p, and Nz - θ_{cmd} - Model Following (Discrete using Step-Response Matrix)

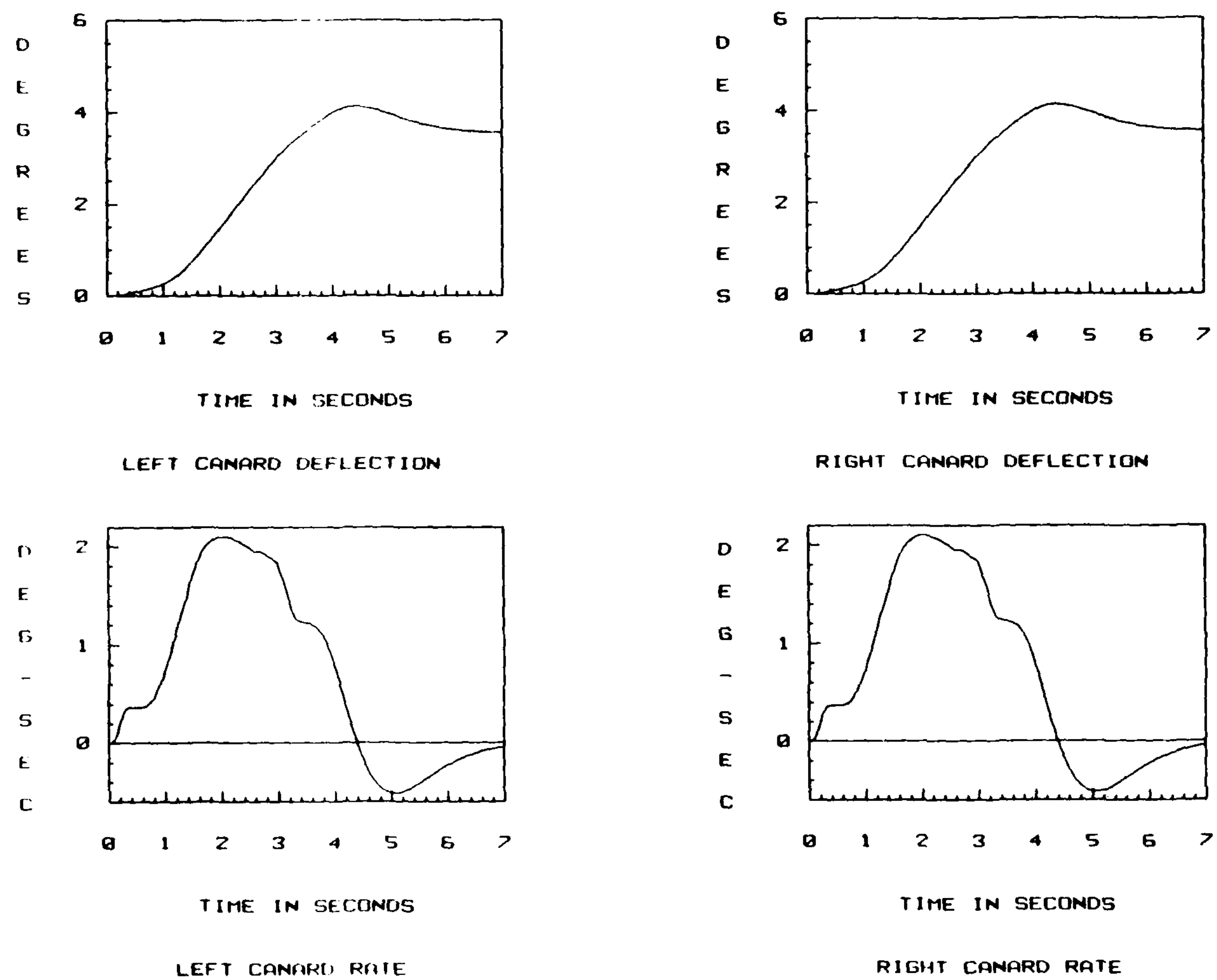


Figure E.62. Canard Deflection and Rates - θ_{cmd} - Model Following (Discrete using Step-Response Matrix)

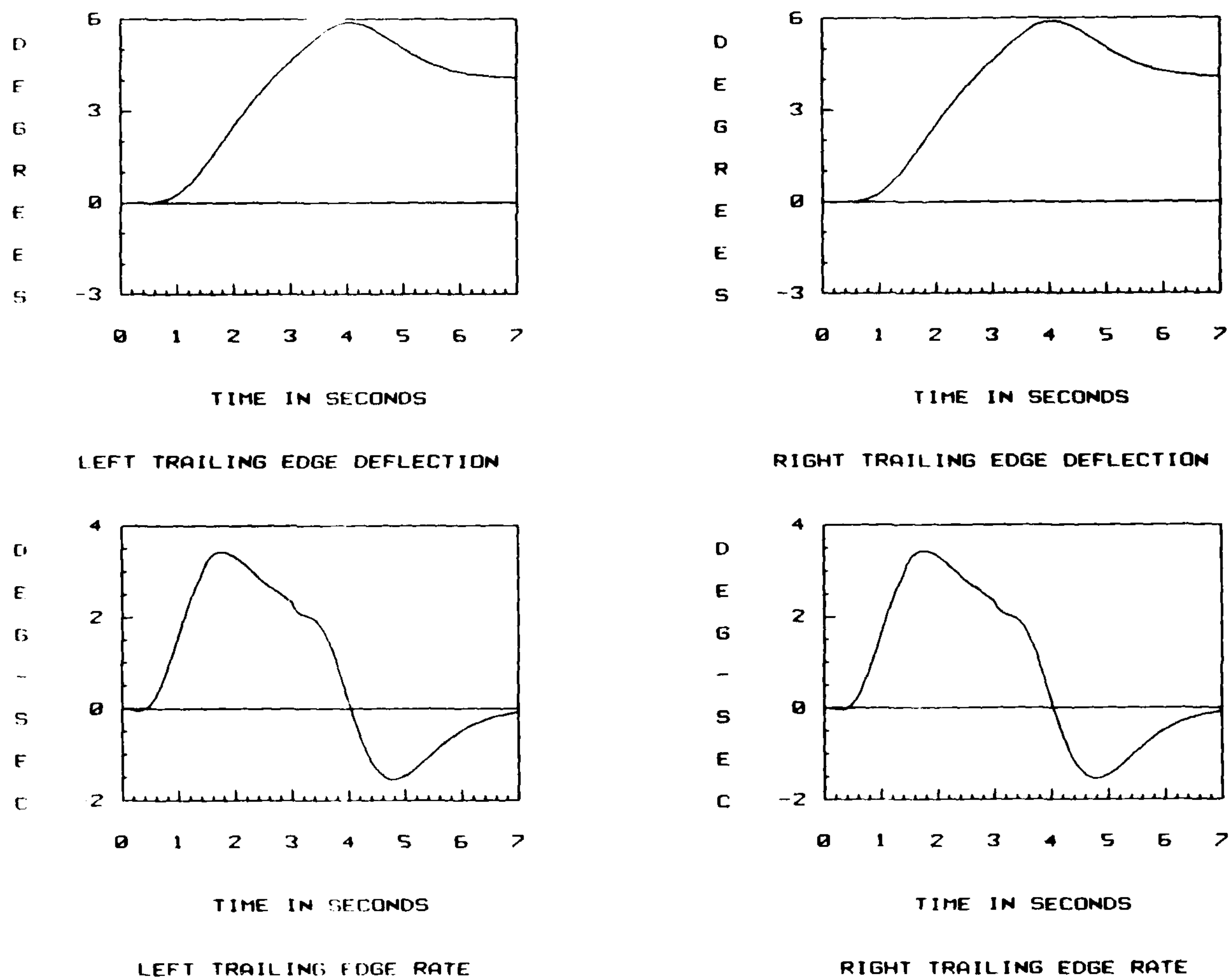


Figure E.63. Trailing Edge Deflection and Rates - θ_{cmd} - Model Following (Discrete using Step-Response Matrix)

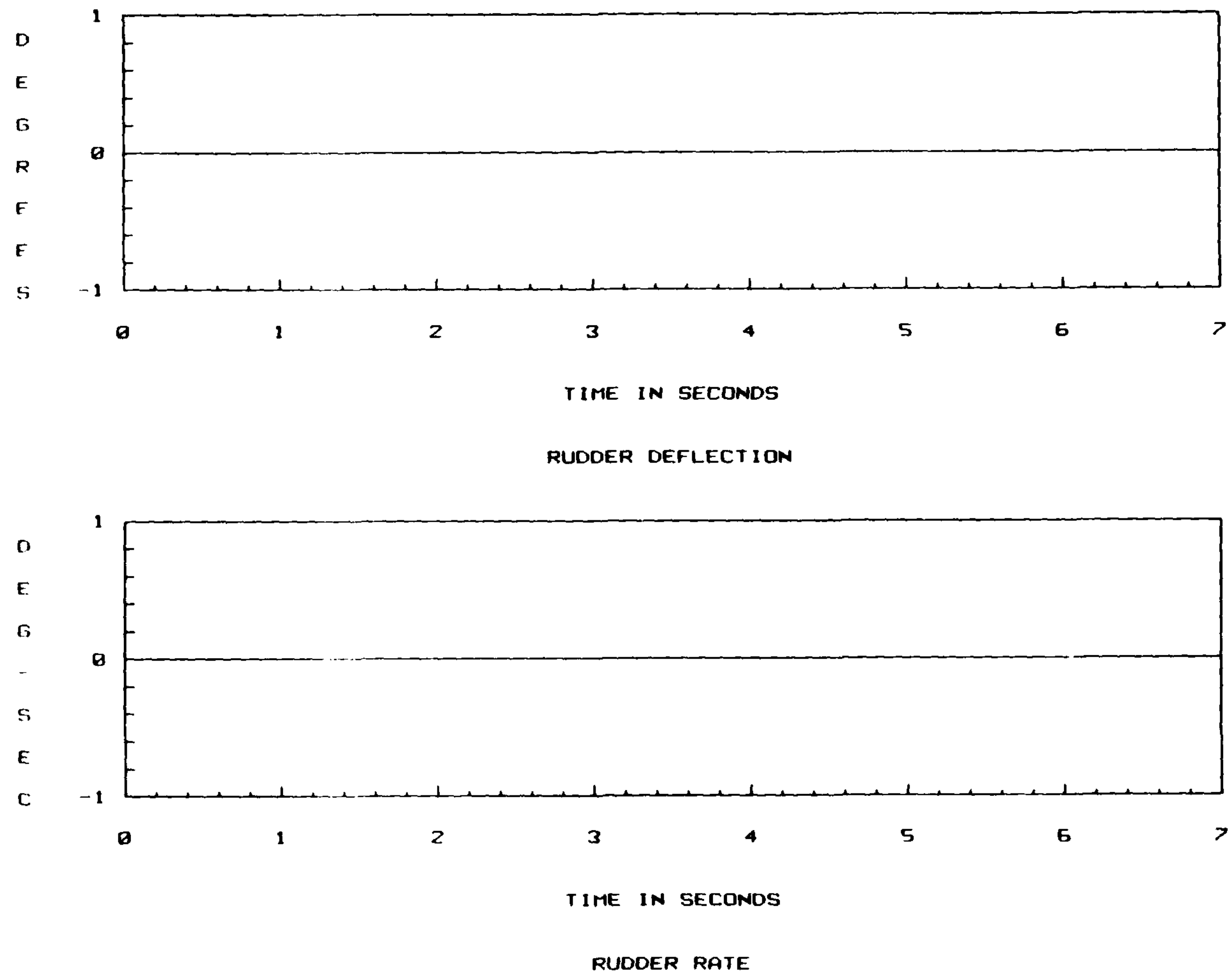


Figure E.64. Rudder Deflection and Rate - θ_{cmd} - Model Following (Discrete using Step-Response Matrix)

Appendix F. *MATRIX$_x$ Simulation Macros*

Introduction

This appendix contains the MATRIX$_x$ macros used in the simulation of the aircraft control laws. The executable file ARMA is described in Appendix B to complement the discussion developing the ARMA plant representation. Generally, the listings are complete with comments interleaved into the executable code. Each of the macros can be executed if typed in as listed and at the MATRIX$_x$ prompt typing EXEC('Filename'). The macro labeled MULTI is the main program and calls several of the other files used in the building of the system. Using executable code is convenient, easy to debug, and takes less storage space than regular MATRIX$_x$ storage files.

Plant Matrices

Once obtained from the LEMCO program of Appendix A, the A and B matrices are provided to MATRIX$_x$ by the following macros. In addition to the standard A and B matrices, the C matrix is expanded to include other outputs, allowing observation of all states and the normal acceleration at the pilot station. The normal acceleration relationships are given in Chapter 2.

```
//                          ACMENTRY PLANT MATRICES

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

 A11 =<

    0.     0.;
    0.     0. >;

 A12        = <

   0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000    0.0000    1.0000    0.0349 >;

 A21        = <

 -32.1804    0.0000;
  -1.0634    0.0000;
   0.0000    0.0000;
   0.0000    0.0360;
   0.0000    0.0000;
   0.0000    0.0000>;

 A22        = <

  -0.0119   -0.0186  -31.2350    0.0000    0.0000    0.0000;
  -0.0324   -1.0634  894.4548    0.0000    0.0000    0.0000;
  -0.0002    0.0069   -0.6015    0.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000   -0.0929    0.0349   -0.9994;
   0.0000    0.0000    0.0000  -27.8066   -2.0376    0.4913;
   0.0000    0.0000    0.0000    2.4582   -0.0241   -0.4377>;

//*************************************************************************

 A = <A11 A12; A21 A22>;

//*************************************************************************
//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
```

```
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

 B1BASIC    = <

    0.    0.    0.    0.    0.    0.    0.    0.    0.;
    0.    0.    0.    0.    0.    0.    0.    0.    0.>;

 B2BASIC    = <

   0.0411  0.0411  0.1322  0.0866  0.1322  0.0866  0.1018  0.1018 0.0000;
  -0.3163 -0.3163 -0.9597 -0.6194 -0.9597 -0.6194 -1.0183 -1.0183 0.0000;
   0.1014  0.1014 -0.0284 -0.0215 -0.0284 -0.0215 -0.0200 -0.0200 0.0000;
   0.0003 -0.0003 -0.0002 -0.0001  0.0002  0.0001 -0.0001  0.0001 0.0006;
   0.0762 -0.0762  0.2219  0.2011 -0.2219 -0.2011  0.1109 -0.1109 0.1144;
   0.0486 -0.0486  0.0029  0.0021 -0.0029 -0.0021  0.0021 -0.0021 -.0544>;

 B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
      B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

 B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
      B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//**************************************************************************

 B  = <B1 ; B2>;

 C1          = <

    0.    0.;
    0.    0.;
    1.    0.;
    0.    1.;
    0.    0.>;


 C2          = <

   1.0000    0.0349    0.0000    0.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000    1.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000    0.0000    0.0000    1.0000>;

//**************************************************************************
 C           = <
```

```
 0.0000 0.0000 1.0000 0.0349 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

//CONTROLLED OUTPUTS ARE AS FOLLOWS;
//Y1 = VELOCITY
//Y2 = BETA
//Y3 = THETA
//Y4 = PHI
//Y5 = YAW RATE "R"
//ADDITIONAL OUTPUTS ARE AS FOLLOWS
//Y6 = "W"
//Y7 = "Q" PITCH RATE
//Y8 = "P" ROLL RATE
//Y9 = "NZ" NORMAL ACCELERATION AT CG-AZ DIVIDED BY G WHERE,
//      AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 895.92 0 0 0>+<1.1228 0 0 0 0 0 0 0>)/(-32.174);

//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATIION AT PILOT STATION

MSU   = -.00023492;
MW    =  .006727;
MQ    = -.559;
MWDOT = -.00006742;
MDCL  =  .07153;
MDCR  =  .10753;
MDTEL =  -.0293 - .0224 - .02068;
MDTER =  -.0293 - .0224 - .02068;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;

NZM=<MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MSU MW MWDOT MQ

//***************************************************************************
D          =<
```

```
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000>;

//************************************************************************

 S = <A B;C D>;

//************************************************************************
//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN

 M          = <

   0.0000    0.0000;
   0.0000    0.0000;
   0.2500    0.0000;
   0.0000    0.2500;
   0.0000    0.0000>;

//************************************************************************
```

ACM Entry Plant Matrices - 30 Percent Loss of Effectiveness - TEL

```
//                    ACM30TL PLANT MATRICES
//                 30 PERCENT LEFT TRAILING EDGE

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

 A11 =<

    0.     0.;
    0.     0. >;
```

```
A12        = <

  0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    1.0000    0.0450 >;

A21        = <

-32.1420    0.0000;
 -1.4370    0.0000;
  0.0000    0.0000;
  0.0000    0.0360;
  0.0000    0.0000;
  0.0000    0.0000>;

A22        = <

 -0.0050    0.0550  -39.9760    0.0000    0.0000    0.0000;
 -0.0240   -1.0280  894.1070    0.0000    0.0000    0.0000;
  0.0000    0.0070   -0.6920    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000   -0.0990    0.0450   -0.9990;
  0.0000    0.0000    0.0000  -31.8340   -2.1380    0.5160;
  0.0000    0.0000    0.0000    2.6670   -0.0310   -0.4420>;

//*************************************************************************


 A = <A11 A12; A21 A22>;

//*************************************************************************

//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

 B1BASIC    = <

    0.    0.    0.    0.    0.    0.    0.    0.    0.;
    0.    0.    0.    0.    0.    0.    0.    0.    0.>;

 B2BASIC    = <

  0.0520  0.0520  0.0000  0.0780  0.1040  0.0670  0.0930  0.0800 0.0000;
```

```
-0.3300 -0.3300  0.0000 -0.6220 -0.9650 -0.6220 -0.7350 -0.7350 0.0000;
 0.1020  0.1020  0.0000 -0.0210 -0.0280 -0.0210 -0.0200 -0.0200 0.0000;
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 0.0010;
 0.0800 -0.0760  0.0000  0.2010 -0.2220 -0.2010  0.1110 -0.1110 0.1150;
 0.0490 -0.0480  0.0000  0.0030 -0.0040 -0.0030  0.0030 -0.0030 0.0550>;


B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
     B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
     B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//**********************************************************************

B  = <B1 ; B2>;

C1          = <

   0.    0.;
   0.    0.;
   1.    0.;
   0.    1.;
   0.    0.>;


C2          = <

  1.0000    0.0410    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    1.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    1.0000>;

//**********************************************************************

 C          = <

  0.0000 0.0000 1.0000 0.0410 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
  0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
```

```
 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

//CONTROLLED OUTPUTS ARE AS FOLLOWS;
//Y1 = VELOCITY
//Y2 = BETA
//Y3 = THETA
//Y4 = PHI
//Y5 = YAW RATE "R"
//ADDITIONAL OUTPUTS ARE AS FOLLOWS
//Y6 = "W"
//Y7 = "Q" PITCH RATE
//Y8 = "P" ROLL RATE
//Y9 = "NZ" NORMAL ACCELERATION AT CG-AZ DIVIDED BY G WHERE,
//      AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 895.92 0 0 0>+<1.4370 0 0 0 0 0 0 0>)/(-32.174);


//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATIION AT PILOT STATION

MSU   = -.00031100;
MW    =  .006950;
MQ    = -.6289;
MWDOT = -.00007020;
MDCL  =  .10000;
MDCR  =  .10000;
MDTEL =  0.0000 - .0214 - .01982;
MDTER =  -.0283 - .0214 - .01982;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;
NZM=<MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MSU MW MD MWDOT MQ

//*************************************************************************


 D          =<

  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
```

```
 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000>;

//**************************************************************************

S = <A B;C D>;

//**************************************************************************

//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN
M          = <

  0.0000    0.0000;
  0.0000    0.0000;
  0.2500    0.0000;
  0.0000    0.2500;
  0.0000    0.0000>;
```

<u>ACM Entry Plant Matrices - 50 Percent Loss of Effectiveness - CL</u>

```
//                      ACM50CL PLANT MATRICES
//                  50 PERCENT LEFT CANARD FAILURE

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

A11 =<

   0.    0.;
   0.    0. >;

A12        = <

  0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
```

```
  0.0000    0.0000    0.0000    0.0000    1.0000    0.0390 >;

A21         = <

-32.1500    0.0000;
 -1.2460    0.0000;
  0.0000    0.0000;
  0.0000    0.0360;
  0.0000    0.0000;
  0.0000    0.0000>;

A22         = <

 -0.0080    0.0580  -36.6690    0.0000    0.0000    0.0000;
 -0.0290   -1.0450  894.3280    0.0000    0.0000    0.0000;
  0.0000    0.0040   -0.6460    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000   -0.0820    0.0390   -0.9990;
  0.0000    0.0000    0.0000  -23.8380   -2.0700    0.4990;
  0.0000    0.0000    0.0000    2.4580   -0.0270   -0.4520>;

//**************************************************************************

A = <A11 A12; A21 A22>;

//**************************************************************************
//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

B1BASIC     = <

   0.    0.    0.    0.    0.    0.    0.    0.    0.;
   0.    0.    0.    0.    0.    0.    0.    0.    0.>;

B2BASIC     = <

-0.1560 -0.1150  0.1320  0.0840  0.1320  0.0840  0.1000  0.1000 0.0000;
-0.1150 -0.2250 -0.9610 -0.6210 -0.9610 -0.6210 -0.7320 -0.7320 0.0000;
 0.0360  0.0720 -0.0280 -0.0220 -0.0280 -0.0220 -0.0200 -0.0200 0.0000;
 0.0000 -0.0010  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 0.0010;
 0.0350 -0.0690  0.2220  0.2010 -0.2220 -0.2010  0.1110 -0.1110 0.1140;
 0.0240 -0.0490  0.0030  0.0030 -0.0030 -0.0030  0.0020 -0.0020 -.0550>;
```

```
B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
     B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
     B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//**************************************************************************

B  = <B1 ; B2>;

C1          = <

   0.    0.;
   0.    0.;
   1.    0.;
   0.    1.;
   0.    0.>;


C2          = <

  1.0000    0.0349    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    1.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    1.0000>;
//*************************************************************************
 C          = <

  0.0000 0.0000 1.0000 0.0349 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
  0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

 //CONTROLLED OUTPUTS ARE AS FOLLOWS;
 //Y1 = VELOCITY
 //Y2 = BETA
 //Y3 = THETA
 //Y4 = PHI
```

```
 //Y5 = YAW RATE "R"
 //ADDITIONAL OUTPUTS ARE AS FOLLOWS
 //Y6 = "W"
 //Y7 = "Q" PITCH RATE
 //Y8 = "P" ROLL RATE
 //Y9 = "NZ" NORMAL ACCELERATION AT CG-AZ DIVIDED BY G WHERE,
 //       AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 895.92 0 0 0>+<1.2460 0 0 0 0 0 0 0>)/(-32.174);

//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATIION AT PILOT STATION

MSU   = -.00016420;
MW    =  .004190;
MQ    = -.587;
MWDOT = -.00006557;
MDCL  =  .03600;
MDCR  =  .07200;
MDTEL =  -.0284 - .0215 - .01980;
MDTER =  -.0284 - .0215 - .01980;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;
NZM=<MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MSU MW MD MWDOT MQ

//**********************************************************************
 D          =<

  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000>;

//*********************************************************************

 S = <A B;C D>;

//**********************************************************************
```

```
//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN

M          = <

  0.0000    0.0000;
  0.0000    0.0000;
  0.2500    0.0000;
  0.0000    0.2500;
  0.0000    0.0000>;
```

ACM Exit Plant Matrices - No Failures

```
//                    ACMEXIT PLANT MATRICES

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

A11 =<

   0.     0.;
   0.     0. >;

A12        = <

  0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    1.0000    0.5184 >;

A21        = <

-28.5877    0.0000;
 -4.9465   26.9479;
  0.0000    0.0000;
  0.0472    0.0322;
  0.0000    0.0000;
  0.0000    0.0000>;

A22        = <

 -0.2762   -0.0283  -136.2191    0.0000    0.0000    0.0000;
  0.1330   -0.6753   262.7934    0.0000    0.0000    0.0000;
```

```
  -0.0018   0.0035   -0.6511    0.0000    0.0000    0.0000;
   0.0000   0.0000    0.0000   -0.0245    0.4602   -0.8878;
   0.0000   0.0000    0.0000   -7.7280   -1.4530    0.9687;
   0.0000   0.0000    0.0000    -.0889   -0.0543    0.0456>;

//**************************************************************************

 A = <A11 A12; A21 A22>;

//**************************************************************************
//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

 B1BASIC    = <

    0.    0.    0.    0.    0.    0.    0.    0.    0.;
    0.    0.    0.    0.    0.    0.    0.    0.    0.>;

 B2BASIC    = <

  -0.0253 -0.0281  0.0206  0.0131  0.0206  0.0131  0.0154  0.0154 0.0000;
  -0.1512 -0.1471 -0.1302 -0.0848 -0.1302 -0.0848 -0.0997 -0.0997 0.0000;
   0.0161  0.0156 -0.0024 -0.0018 -0.0024 -0.0018 -0.0016 -0.0016 0.0000;
   0.0007 -0.0007 -0.0001 -0.0001  0.0001  0.0001 -0.0001  0.0001 0.0004;
   0.0113 -0.0337  0.0270  0.0240 -0.0270 -0.0240  0.0135 -0.0135 0.0277;
   0.0133 -0.0121 -0.0005 -0.0004  0.0005  0.0004 -0.0004  0.0004 -.0130>;

 B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
      B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

 B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
      B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//***************************************************************************

 B  = <B1 ; B2>;

 C1         = <

    0.    0.;
    0.    0.;
    1.    0.;
    0.    1.;
```

```
   0.    0.>;

C2          = <

 1.0000    0.4600    0.0000    0.0000    0.0000    0.0000;
 0.0000    0.0000    0.0000    1.0000    0.0000    0.0000;
 0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
 0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
 0.0000    0.0000    0.0000    0.0000    0.0000    1.0000>;

//**************************************************************************
C           = <

 0.0000 0.0000 1.0000 0.4600 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

 //CONTROLLED OUTPUTS ARE AS FOLLOWS;
 //Y1 = VELOCITY
 //Y2 = BETA
 //Y3 = THETA
 //Y4 = PHI
 //Y5 = YAW RATE "R"
 //ADDITIONAL OUTPUTS ARE AS FOLLOWS
 //Y6 = "W"
 //Y7 = "Q" PITCH RATE
 //Y8 = "P" ROLL RATE
 //Y9 = "NZ" NORMAL ACCELERATION -AZ DIVIDED BY G WHERE,
 //      AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 296.29 0 0 0>+<14.8015 0 0 0 0 0 0 0>)/(-32.174);

//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATION AT PILOT STATION

MSU    = -.00179226;
MW     =  .003459;
MQ     = -.6511;
MWDOT  = -.001951;
```

```
MDCL   =  .01608;
MDCR   =  .01559;
MDTEL  = -.00238 - .00179 - .00164;
MDTER  = -.00238 - .00179 - .00164;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;

NZM    = <MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MSU MW MQ MWDOT

//************************************************************************
D          =<

  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000>;

//************************************************************************

S = <A B;C D>;

//************************************************************************
//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN

M          = <

  0.0000    0.0000;
  0.0000    0.0000;
  0.2500    0.0000;
  0.0000    0.2500;
  0.0000    0.0000>;

//************************************************************************
```

TF/TA Plant Matrices - No Failures

```
//                          TFTA PLANT MATRICES

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

 A11 =<

   0.    0.;
   0.    0. >;

 A12         = <

  0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    1.0000    0.0153 >;

 A21         = <

 -32.1961    0.0000;
  -0.5002    0.0000;
   0.0000    0.0000;
   0.0000    0.0320;
   0.0000    0.0000;
   0.0000    0.0000>;

 A22         = <

  -0.0355    0.0357   -15.6105    0.0000    0.0000    0.0000;
  -0.0071   -3.2056 -1004.8788    0.0000    0.0000    0.0000;
  -0.0003    0.0202    -1.6773    0.0000    0.0000    0.0000;
   0.0000    0.0000     0.0000   -0.2538    0.0155   -0.9999;
   0.0000    0.0000     0.0000  -66.9300   -5.4612    1.3049;
   0.0000    0.0000     0.0000    8.2821   -0.0299   -1.2709>;

//**********************************************************************

 A = <A11 A12; A21 A22>;

//**********************************************************************
//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

 B1BASIC     = <
```

```
    0.     0.     0.     0.     0.     0.     0.     0.     0.;
    0.     0.     0.     0.     0.     0.     0.     0.     0.>;

B2BASIC    = <

 -0.3284 -0.3284  0.2548  0.1679  0.2548  0.1679  0.1975  0.1975 0.0000;
 -0.6788 -0.6788 -1.7517 -1.1313 -1.7517 -1.1313 -1.3351 -1.3351 0.0000;
  0.2387  0.2387 -0.0534 -0.0406 -0.0534 -0.0406 -0.0372 -0.0372 0.0000;
  0.0012 -0.0012 -0.0001 -0.0001  0.0001  0.0001 -0.0001  0.0003 0.0018;
  0.2336 -0.2336  0.4200  0.3737 -0.4200 -0.3737  0.2100 -0.2100 0.3737;
  0.1590 -0.1590  0.0024  0.0012 -0.0024 -0.0012  0.0012 -0.0012 -.1795>;

B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
     B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
     B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//**********************************************************************

B  = <B1 ; B2>;

C1         = <

    0.    0.;
    0.    0.;
    1.    0.;
    0.    1.;
    0.    0.>;


C2         = <

   1.0000     0.0138     0.0000     0.0000     0.0000     0.0000;
   0.0000     0.0000     0.0000     1.0000     0.0000     0.0000;
   0.0000     0.0000     0.0000     0.0000     0.0000     0.0000;
   0.0000     0.0000     0.0000     0.0000     0.0000     0.0000;
   0.0000     0.0000     0.0000     0.0000     0.0000     1.0000>;

//**********************************************************************
C          = <

   0.0000 0.0000 1.0000 0.0138 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
```

```
 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

//CONTROLLED OUTPUTS ARE AS FOLLOWS;
//Y1 = VELOCITY
//Y2 = BETA
//Y3 = THETA
//Y4 = PHI
//Y5 = YAW RATE "R"
//ADDITIONAL OUTPUTS ARE AS FOLLOWS
//Y6 = "W"
//Y7 = "Q" PITCH RATE
//Y8 = "P" ROLL RATE
//Y9 = "NZ" NORMAL ACCELERATION -AZ DIVIDED BY G WHERE,
//      AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 1005.0 0 0 0>+<.4436 0 0 0 0 0 0 0>)/(-32.174);
//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATION AT PILOT STATION

MSU   = -.0002745;
MW    =  .0199;
MQ    =-1.653359;
MWDOT = -.00016449;
MDCL  =  .2353;
MDCR  =  .2353;
MDTEL = -.05267 - .04 - .0366;
MDTER = -.05267 - .04 - .0366;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;

NZM=<MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MWDOT MQ MW

//*************************************************************************
D          =<

  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
```

```
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000>;

//***************************************************************************

 S = <A B;C D>;

//***************************************************************************
//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN

 M          = <

   0.0000    0.0000;
   0.0000    0.0000;
   0.2500    0.0000;
   0.0000    0.2500;
   0.0000    0.0000>;

//************************************************************************
```

TF/TA Plant Matrices - 30 Percent Loss of Effectiveness - TEL

```
//                      TFTA30TL.DAT PLANT MATRICES
//                       (100 PERCENT LOE TE1L)

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

 A11 =<

    0.    0.;
    0.    0. >;

 A12        = <
```

```
  0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    1.0000    0.0160>;

A21         = <

-32.1704    0.0000;
 -0.5110    0.0000;
  0.0000    0.0000;
  0.0000    0.0320;
  0.0000    0.0000;
  0.0000    0.0000>;

A22         = <

 -0.0330   -0.0200  -15.2961    0.0000    0.0000    0.0000;
 -0.0080   -3.0620 1004.8730    0.0000    0.0000    0.0000;
  0.0000    0.0210   -1.8240    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000   -0.2460    0.0160   -1.0000;
  0.0000    0.0000    0.0000  -65.9740   -5.4310    1.2860;
  0.0000    0.0000    0.0000    8.1650   -0.0340   -1.2630>;

//***********************************************************************

A = <A11 A12; A21 A22>;

//***********************************************************************
//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

B1BASIC     = <

   0.    0.    0.    0.    0.    0.    0.    0.    0.;
   0.    0.    0.    0.    0.    0.    0.    0.    0.>;

B2BASIC     = <

-0.3450 -0.3450  0.0930  0.1580  0.2520  0.1580  0.1940  0.1940 0.0000;
-0.6760 -0.6760  0.0000 -1.1140 -1.7250 -1.1140 -1.3160 -1.3160 0.0000;
 0.2360  0.2360  0.0000 -0.0400 -0.0520 -0.0400 -0.0360 -0.0360 0.0000;
 0.0010 -0.0010 -0.0000 -0.0000  0.0000  0.0000 -0.0000  0.0000 0.0020;
 0.2300 -0.2300  0.0000  0.3680 -0.4150 -0.3680  0.2070 -0.2070 0.3800;
 0.1570 -0.1570  0.0000  0.0010 -0.0020 -0.0010  0.0010 -0.0010 -.1770>;
```

```
B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
     B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
     B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//**********************************************************************
B   = <B1 ; B2>;

C1          = <

   0.    0.;
   0.    0.;
   1.    0.;
   0.    1.;
   0.    0.>;


C2          = <

  1.0000    0.0160    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    1.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    1.0000>;

//**********************************************************************

C           = <

  0.0000 0.0000 1.0000 0.0160 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
  0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

  //CONTROLLED OUTPUTS ARE AS FOLLOWS;
  //Y1 = VELOCITY
  //Y2 = BETA
  //Y3 = THETA
```

```
//Y4 = PHI
//Y5 = YAW RATE "R"
//ADDITIONAL OUTPUTS ARE AS FOLLOWS
//Y6 = "W"
//Y7 = "Q" PITCH RATE
//Y8 = "P" ROLL RATE
//Y9 = "NZ" NORMAL ACCELERATION -AZ DIVIDED BY G WHERE,
//      AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 1005 0 0 0>+<0.51090 0 0 0 0 0 0 0>)/(-32.174);


//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATIION AT PILOT STATION

MSU   =  -.00032138;
MW    =   .020200;
MQ    = -1.6586;
MWDOT = -.00016501;
MDCL  =  .23586;
MDCR  =  .23586;
MDTEL =  -.0366 - .0000 - .04000;
MDTER =  -.0366 - .0256 - .04000;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;

NZM=<MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MSU MW MD MWDOT MQ

//**********************************************************************
 D          =<

  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000>;

//**********************************************************************
S = <A B;C D>;
```

```
//**********************************************************************
//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN

 M          = <

   0.0000    0.0000;
   0.0000    0.0000;
   0.2500    0.0000;
   0.0000    0.2500;
   0.0000    0.0000>;
```

TF/TA Plant Matrices - 50 Percent Loss of Effectiveness - CL

```
//                     TFTA50CL.DAT PLANT MATRICES
//                     (50 PERCENT LOE CANARD LEFT)

//THE FOLLOWING LISTINGS OF MATRICES CAN BE PUT INTO MATRIXX BY
//COPYING THE FOLLOWING PROGRAM INTO A <FILENAME>.DAT AND
//USING THE EXEC('<FILENAME>') COMMAND IN MATRIXX

 A11 =<

    0.     0.;
    0.     0. >;

 A12        = <

   0.0000    0.0000    1.0000    0.0000    0.0000    0.0000;
   0.0000    0.0000    0.0000    0.0000    1.0000    0.0110>;

 A21        = <

 -32.1720    0.0000;
  -0.3540    0.0000;
   0.0000    0.0000;
   0.0000    0.0320;
   0.0000    0.0000;
   0.0000    0.0000>;

 A22        = <
```

```
 -0.0380   -0.0290  -11.0560    0.0000    0.0000    0.0000;
 -0.0160   -3.6430 1005.4390    0.0000    0.0000    0.0000;
  0.0000    0.0140   -2.1070    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000   -0.2780    0.0110   -1.0000;
  0.0000    0.0000    0.0000  -77.1970   -6.1840    1.4760;
  0.0000    0.0000    0.0000    8.7570   -0.0290   -1.4650>;

//*********************************************************************

 A = <A11 A12; A21 A22>;

//*********************************************************************
//THE COMPLETE 9 DEGREE OF FREEDOM BMATRIX IS LISTED BELOW
//SINCE THE AIRCRAFT TRAILING EDGE SURFACES WERE TIED TOGETHER
//COLUMNS 3,4 AND 7 WERE COMBINED AND 5,6,AND 8 WERE COMBINED
//LEAVING A 5 INPUT 5 OUTPUT SYSTEM FOR PORTER ANALYSIS

 B1BASIC    = <

   0.    0.    0.    0.    0.    0.    0.    0.    0.;
   0.    0.    0.    0.    0.    0.    0.    0.    0.>;

 B2BASIC    = <

 -0.5220 -0.3790  0.2940  0.1930  0.2940  0.1930  0.2270  0.2270 0.0000;
 -0.3950 -0.7740 -2.0190 -1.3040 -2.0190 -1.3040 -1.5390 -1.5390 0.0000;
  0.1390  0.2750 -0.0610 -0.0470 -0.0610 -0.0470 -0.0430 -0.0430 0.0000;
  0.0010 -0.0020 -0.0000 -0.0000  0.0000  0.0000 -0.0000  0.0000 0.0020;
  0.1210 -0.2830  0.4850  0.4310 -0.4850 -0.4310  0.2430 -0.2430 0.4310;
  0.0920 -0.1820  0.0010  0.0010 -0.0010 -0.0010  0.0010 -0.0010 -.2070>;

 B1= <B1BASIC(:,1),B1BASIC(:,2),B1BASIC(:,3)+B1BASIC(:,4)+B1BASIC(:,7),...
      B1BASIC(:,5)+B1BASIC(:,6)+B1BASIC(:,8),B1BASIC(:,9)>;

 B2= <B2BASIC(:,1),B2BASIC(:,2),B2BASIC(:,3)+B2BASIC(:,4)+B2BASIC(:,7),...
      B2BASIC(:,5)+B2BASIC(:,6)+B2BASIC(:,8),B2BASIC(:,9)>;

//*********************************************************************
 B  = <B1 ; B2>;

 C1          = <

   0.    0.;
   0.    0.;
   1.    0.;
```

```
   0.    1.;
   0.    0.>;


C2          = <

  1.0000    0.0110    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    1.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000;
  0.0000    0.0000    0.0000    0.0000    0.0000    1.0000>;

//*********************************************************************

C           = <

  0.0000 0.0000 1.0000 0.0110 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000;
  1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000;
  0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000;
  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000>;

 //CONTROLLED OUTPUTS ARE AS FOLLOWS;
 //Y1 = VELOCITY
 //Y2 = BETA
 //Y3 = THETA
 //Y4 = PHI
 //Y5 = YAW RATE "R"
 //ADDITIONAL OUTPUTS ARE AS FOLLOWS
 //Y6 = "W"
 //Y7 = "Q" PITCH RATE
 //Y8 = "P" ROLL RATE
 //Y9 = "NZ" NORMAL ACCELERATION -AZ DIVIDED BY G WHERE,
 //       AZ=WDOT -(UO)Q + (GSTHEO)DELTHETA FT/SEC SQ

C(9,:)=(A(4,:)+<0 0 0 0 1005 0 0 0>+<0.35376 0 0 0 0 0 0 0>)/(-32.174);


//DEFINITIONS FOR THETADD FOR NORMAL ACCELERATION AT PILOT STATION

MSU    =   -.00015109;
```

```
MW     =   .013740;
MQ     = -1.9152;
MWDOT = -.00019046;
MDCL  =  .13933;
MDCR  =  .27531;
MDTEL =  -.0435 - .0162 - .04689;
MDTER =  -.0435 - .0162 - .04689;

C(10,:)=<C(9,:)+(MSU*C(1,:)+MW*C(6,:)+MQ*C(7,:)+MWDOT*A(4,:))/1.15>;

NZM=<MDCL,MDCR,MDTEL,MDTER>;

CLEAR MDCL MDCR MDTEL MDTER MSU MW MD MWDOT MQ

//*********************************************************************
 D          =<

   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000;
   0.0000 0.0000 0.0000 0.0000 0.0000>;

//*********************************************************************
 S = <A B;C D>;

//*********************************************************************
//THE MEASUREMENT MATRIX CHOSEN FOR THE IRREGULAR PLANT DESIGN

 M          = <

   0.0000    0.0000;
   0.0000    0.0000;
   0.2500    0.0000;
   0.0000    0.2500;
   0.0000    0.0000>;
```

F.3 Continuous Time System Analysis Macros

The continuous time analysis of the system with the PI control law is simulated in $MATRIX_x$ using the following macros.

Main Program for CRCA Simulation - Ramped Input

```
//THIS PROGRAM SIMULATES THE SYSTEM WITH ACTUATORS
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';-
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.1;
LGT1=(LGT-.5)/.1;//SETS THE RAMPING FOR SURFACE LIMITS
T1=<0:.1:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=.03491*<0:.1:3.0>';
    R2B=<R2A;3*.03491*ONES(LGT2,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
    R8A =2*.7854*<0:.1:.5>';
    R8A1=2*.02542*<0:.1:.5>';
    R8B =<R8A;.7854*ONES(LGT1,1)>;
    R8B1=<R8A1;.02542*ONES(LGT1,1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),R8B,R8B1>;
    R10A=2*.0349*<0:.1:.5>';
    R10B=<R10A;.0349*ONES(LGT1,1)>;
R10=<0*ONES(T1),R10B,0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B R8A R8A1 R8B R8B1 R10A R10B
//
INQUIRE G1 'WHAT IS YOUR GAIN (G) ESTIMATE: '
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
INQUIRE L 'WHAT IS YOUR LAMBDA 0 MULTIPLIER: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
M(3,1)=KQ;M(4,2)=KP;
SIGMA=<SIG1,0,0,0,0;0,SIG2,0,0,0;0,0,SIG3,0,0;0,0,0,SIG4,0;...
       0,0,0,0,SIG5>;
TITLE1='THE FOLLOWING ARE THE PARAMETERS USED IN THE SIMULATION';
DISPLAY(TITLE1)
G=G1*EYE(5)
SIGMA
F2=C2+M*A12;
```

```
K1=INV(F2*B2)*SIGMA
K2=L*K1
EXEC('M12');//ACTUATOR BUILD
EXEC('M8');//PI CONTROLLER BUILD
EXEC('M7');//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6');//PLANT MATRIX WITH ACTUATORS
EXEC('M11');//MULTI SUPERBLOCK WITH ACTUATORS
EXEC('M19');//PRE FILTERS BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH PREFILTERS
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Main Program for CRCA Simulation - Model Following Input

```
//MULTI1.DAT - SIMULATION WITH MODEL FOLLOWING
//SIMULATION WITH ACTUATORS
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.1;
T1=<0:.1:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=.03491*<0:.1:3.0>';
    R2B=<R2A;3*.03491*ONES(LGT2,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),.7854*ONES(T1),.02542*ONES(T1)>;
R10=<0*ONES(T1),.0349*ONES(T1),0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B
//
INQUIRE G1 'WHAT IS YOUR GAIN (G) ESTIMATE: '
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
INQUIRE L 'WHAT IS YOUR LAMBDA 0 MULTIPLIER: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
M(3,1)=KQ;M(4,2)=KP;
```

```
SIGMA=<SIG1,0,0,0,0;0,SIG2,0,0,0;0,0,SIG3,0,0;0,0,0,SIG4,0;...
       0,0,0,0,SIG5>;
TITLE1='THE FOLLOWING ARE THE PARAMETERS USED IN THE SIMULATION';
DISPLAY(TITLE1)
G=G1*EYE(5)
SIGMA
F2=C2+M*A12;
K1=INV(F2*B2)*SIGMA
K2=L*K1
EXEC('M12');//ACUATOR BUILD
EXEC('M8') ;//PI CONTROLLER
EXEC('M7') ;//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6') ;//PLANT WITH ACTUATORS SUPER BLOCK
EXEC('M11');//MULTI SUPERBLOCK WITH ACTUATORS
EXEC('M15');//MODEL FOLLOWING BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH MODEL FOLLOWING
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Aircraft Model with Actuators

```
//M6.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A SYSTEM MATRIX SUPER BLOCK WITH ACTUATORS
//MAKE SURE THE S MATRIX IS DEFINED
//ADDS PLANT MATRIX
PLANT1
0
DEF,2, SUPER,  NAME,PLANT,        INPUTS,5, OUT,12,ENTRY
DEF,1, SUPER,  NAME,ACT,          INPUTS,5, OUT,10,ENTRY
CONNECT
INPUT,5,1,Y,    INTERNAL,1,2,2,1,4,2,6,3,8,4,10,5,0,0

OUTPUT,22,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUTPUT,22,1,1,13,2,14,3,15,4,16,5,17,6,18,7,19,8,20,9,21,10,22,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Aircraft Model without Actuators

```
//M7.DAT
BUILD
PLOT OFF
EDIT
PLANT
0
DEF,1, DYN,       STATE, NAME,B,    INPUTS,5,  OUT,8,STATES,0,ENTRY,B
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1, INPUTS,2,  OUT,8,ENTRY,2,1,1
DEF,3, DYN,       NTH,   NAME,INT,
OUT,8,ENTRY,1,1,1,1,1,1,1,1,1,Y
DEF,4, DYN,       STATE, NAME,C,    INPUTS,8,  OUT,10,STATES,0,ENTRY,C
DEF,5, DYN,       STATE, NAME,A,    INPUTS,8,  OUT,8,STATES,0,ENTRY,A
CONNECT
INPUT,5,1,Y,      INTERNAL,1,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,0,0
                  INTERNAL,2,3,Y
                  INTERNAL,3,4,Y
                  INTERNAL,3,5,Y
                  INTERNAL,5,2,1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,0,0
OUTPUT,12,4,1,1,2,2,3,3,4,4,5,5,6,8,7,9,8,10,9,11,10,12,0,0
OUTPUT,12,2,1,6,2,7,0,0
//OUTPUT FROM BLOCK 2 - 6 AND 7 ARE XDOT(A11 AND A12)
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

PI Controller

```
//M8.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A PI CONTROLLER
//THE CURRENT INPUTS ARE 5 WITH K1 AND K2
//DIMENSIONED 5 X 5
//TO CHANGE THEM FOR A 3 INPUT SYSTEM JUST
//CHANGE OUTPUTS AND INPUTS FROM 5 TO WHATEVER YOU WANT
PI
0
DEF,1, DYN,       Nth,  NAME,INT,                OUT,5,ENTRY,1,1,1,1,1,1,YES
DEF,2, DYN,       STATE,NAME,K2,    INPUTS,5,  OUT,5,STATES,0,ENTRY,K2
DEF,5, DYN,       STATE,NAME,K1,    INPUTS,5,  OUT,5,STATES,0,ENTRY,K1
```

```
DEF,3, ALGEBRAIC, SUM,  NAME,SUM1, INPUTS,2,  OUT,5,ENTRY,2,1,1
CONNECT
INPUT,5,1,Y,             INTERNAL,1,2,Y,           OUTPUT,5,3,Y
INPUT,5,5,Y,             INTERNAL,2,3,1,1,2,2,3,3,4,4,5,5,0,0
                         INTERNAL,5,3,1,6,2,7,3,8,4,9,5,10,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Plant and Controller

```
//M11.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A MULTI SUPERBLOCK FOR SIMULATION - 5 INPUTS
//INSURE YOU HAVE CREATED THE SUPERBLOCKS PI AND APPROPRIATE PLANT
//THIS IS A SIMULATION FOR AN IRREGULAR PLANT WITH ACTUATORS
//THIS SUPERBLOCK WILL HAVE 10 OUTPUTS- OUTPUT 1-5 WILL BE COMMAND INPUTS
//OUTPUTS 5-14 WILL BE STATE VARIABLES
MULTI
0
DEF,2, SUPER,           NAME,PI,     INPUTS,5,     OUT,5,          ENTRY
DEF,4, SUPER,           NAME,PLANT1,INPUTS,5,      OUT,22,         ENTRY
DEF,3, DYN,      STATE,NAME,G,       INPUTS,5,     OUT,5,STATES,0,ENTRY,G
DEF,5, DYN,      STATE,NAME,M,       INPUTS,2,     OUT,5,STATES,0,ENTRY,M
DEF,1, ALGEBRAIC, SUM,  NAME,SUM1,   INPUTS,2,     OUT,5,ENTRY,2,1,-1
DEF,6, ALGEBRAIC, SUM,  NAME,W,      INPUTS,2,     OUT,5,ENTRY,2,1,1
CONNECT
INPUT,5,1,1,1,2,2,3,3,4,4,5,5,0,0,
                              INTERNAL,1,2,Y
                              INTERNAL,2,3,Y
                              INTERNAL,3,4,Y
                              INTERNAL,4,5,6,1,7,2,0,0
                              INTERNAL,4,6,1,1,2,2,3,3,4,4,5,5,0,0
                              INTERNAL,5,6,1,6,2,7,3,8,4,9,5,10,0,0
                              INTERNAL,6,1,1,6,2,7,3,8,4,9,5,10,0,0
OUTPUT,20,4,14,1,16,2,18,3,20,4,22,5,0,0
OUTPUT,20,4,13,16,15,17,17,18,19,19,21,20,0,0
OUTPUT,20,4,1,6,2,7,3,8,4,9,5,10,8,11,9,12,10,13,11,14,12,15,0,0

//OUTPUTS 1-5 ARE CONTROL OUTPUTS:CL,CR,TEL,TER,RUDDER
```

```
//OUTPUTS 6-15 ARE STATE OUTPUTS:VEL,THETA,BETA,PHI,R,W,Q,P,NZ,NZPILOT
//OUTPUTS 13,15,17,19,21 RATES FOR ACTUATORS CL,CR,TEL,TER,RUD
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Actuators

```
//M12.DAT
//THIS ROUTINE BUILDS THE ACTUATORS FOR THE CRCA
//FIRST ORDER MODEL OF XD=-20X+20U
BACT=20;//B MATRIX FOR THE ACTUATOR
AACT=-20;//A MATRIX FOR THE ACTUATOR
BUILD
PLOT OFF
EDIT
ACL
0
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,   OUT,1,ENTRY,30,-60,1,Y
DEF,5, DYN,       STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,               INTERNAL,1,2,1,1,0,0
                         INTERNAL,2,3
                         INTERNAL,3,4
                         INTERNAL,4,5
                         INTERNAL,5,2,1,2,0,0
OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING THE RIGHT CANARD ACTUATOR
COPY,       ACL,        ACR

//BUILDING TRAILING EDGE LEFT ACTUATOR
EDIT
ATL1
0
```

```
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,   OUT,1,ENTRY,30,-30,1,Y
DEF,5, DYN,       STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,               INTERNAL,1,2,1,1,0,0
                         INTERNAL,2,3
                         INTERNAL,3,4
                         INTERNAL,4,5
                         INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING TRAILING EDGE RIGHT ACTUATOR
COPY,        ATL1,      ATR1

//BUILDING RUDDER ACTUATOR
EDIT
ATRUD
0
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,   OUT,1,ENTRY,20,-20,1,Y
DEF,5, DYN,       STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,               INTERNAL,1,2,1,1,0,0
                         INTERNAL,2,3
                         INTERNAL,3,4
                         INTERNAL,4,5
                         INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILD SUPER BLOCK OF ACTUATORS
EDIT
ACT
0
DEF, 1, SUPER,            NAME,ACL,   INPUTS,1, OUT,2,ENTRY
DEF, 2, SUPER,            NAME,ACR,   INPUTS,1, OUT,2,ENTRY
```

```
DEF, 3, SUPER,           NAME,ATL1,  INPUTS,1, OUT,2,ENTRY
DEF, 4, SUPER,           NAME,ATR1,  INPUTS,1, OUT,2,ENTRY
DEF, 5, SUPER,           NAME,ATRUD, INPUTS,1, OUT,2,ENTRY
CONNECT
INPUT,5,1,1,1,0,0,                   OUTPUT,10,1,1,1,2,2,0,0
INPUT,5,2,2,1,0,0,                   OUTPUT,10,2,1,3,2,4,0,0
INPUT,5,3,3,1,0,0,                   OUTPUT,10,3,1,5,2,6,0,0
INPUT,5,4,4,1,0,0,                   OUTPUT,10,4,1,7,2,8,0,0
INPUT,5,5,5,1,0,0,                   OUTPUT,10,5,1,9,2,10,0,0

TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR BACT AACT
```

Model Following Filter

```
//M15.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A MODEL FOLLOWING PROGRAM
PRFB=<-12,-57,-100,1;1,0,0,0;0,1,0,0;0,0,100,0>;
PRFR=PRFB;
PRFP=<-18,-105,-250,1;1,0,0,0;0,1,0,0;0,0,250,0>;
PRFV=PRFP;
PRFT=<-10,-41,-50,1;1,0,0,0;0,1,0,0;0,0,50,0>;
BUILD
PLOT OFF
EDIT
PREF
0
DEF,1, DYN,  STATE,NAME,  PRE1,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFV,Y
DEF,2, DYN,  STATE,NAME,  PRE2,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFB,Y
DEF,3, DYN,  STATE,NAME,  PRE3,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFT,Y
DEF,4, DYN,  STATE,NAME,  PRE4,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFP,Y
DEF,5, DYN,  STATE,NAME,  PRE5,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFR,Y
CONNECT
INPUT,5,1,1,1,0,0,                          OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                          OUT,5,2,1,2,0,0
INPUT,5,3,3,1,0,0,                          OUT,5,3,1,3,0,0
INPUT,5,4,4,1,0,0,                          OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                          OUT,5,5,1,5,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
```

```
MAT
CLEAR PRFV PRFB PRFT PRFP PRFR
```

Aircraft Model with PI Controller and Pre-Filter

```
//M16.DAT
//THIS EXEC"TAB'" FILE BUILDS A MULTI SUPERBLOCK WITH PRE FILTER
//ALSO BUILDS NORMAL ACCELERATION AT PILOT STATION OUTPUT
BUILD
PLOT OFF
EDIT
MULTI1
0
DEF,2,  SUPER,      NAME,MULTI,        INPUTS,5,   OUT,20,  ENTRY
DEF,1,  SUPER,      NAME,PREF,         INPUTS,5,   OUT,5,   ENTRY
DEF,5,  DYN,STATE,  NAME,NZM,          INPUTS,4,   OUT,1,STATES,0,ENTRY,NZM
DEF,4,  ALGEBRAIC,  SUM,NAME,NZ,       INPUTS,2,   OUT,1,ENTRY,2,1,1
CONNECT
INPUT,5,1,Y,        INTERNAL,1,2,Y
                    INTERNAL,2,5,1,1,2,2,3,3,4,4,0,0
                    INTERNAL,5,4,1,1,0,0
                    INTERNAL,2,4,14,2,0,0
OUT,25,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUT,25,2,13,13,14,14,0,0
OUT,25,4,1,15,0,0
OUT,25,1,1,16,2,17,3,18,4,19,5,20,0,0
OUT,25,2,16,21,17,22,18,23,19,24,20,25,0,0
//OUTPUTS 21,22,23,24,25 ARE RATES FOR CL,CR,TEL,TER,RUD
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Simulation and Plotting

```
//M18.DAT
ERASE;CLEAR Y
INQUIRE TYPE 'FLIGHT MANEUVER R2, R8, OR R10: '
Y=SIM(T1,TYPE);
INQUIRE PLT 'DO YOU WANT HARDCOPIES'
```

```
INQUIRE PLT1 '1= R8 PLOTS -- 0= R2 PLOTS --2= R10 PLOTS';
IF PLT1=1,EXEC('R8')
IF PLT1=0,EXEC('R2')
IF PLT1=2,EXEC('R10')
```

Low Pass Filter

```
//M19.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A LOW PASS PREFILTER
PRF=<-10,1;10,0>;
BUILD
PLOT OFF
EDIT
PREF
0
DEF,1, DYN,  STATE,NAME,  PRE1,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,2, DYN,  STATE,NAME,  PRE2,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,3, DYN,  STATE,NAME,  PRE3,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,4, DYN,  STATE,NAME,  PRE4,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,5, DYN,  STATE,NAME,  PRE5,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
CONNECT
INPUT,5,1,1,1,0,0,                           OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                           OUT,5,2,1,2,0,0
INPUT,5,3,3,1,0,0,                           OUT,5,3,1,3,0,0
INPUT,5,4,4,1,0,0,                           OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                           OUT,5,5,1,5,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR PRF
```

Pitch Rate Tracking Plots

```
//R2.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE PITCH RATE TRACKING
PLOT(T1,Y(:,[8])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          PITCH ANGLE/,UPPER')
PLOT(T1,Y(:,[12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          PITCH RATE/,LOWER,KEEP')
```

```
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,...
    YMIN=-1,YMAX=1,UPPER LEFT');
PLOT(T1,Y(:,[10 13])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        ROLL RATE (P) VS YAW RATE (R)/,...
    YMIN=-1,YMAX=1,LOWER LEFT,KEEP')
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/         VELOCITIES (U) AND (W)/,UPPER RIGHT')
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       LEFT CANARD DEFLECTION/,UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT CANARD RATE/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT CANARD RATE/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT TRAILING EDGE RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          RUDDER DEFLECTION/,...
    YMIN=-1,YMAX=1,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RUDDER RATE/,...
    YMIN=-1,YMAX=1,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,18)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          PITCH RATE COMMAND/,UPPER');
IF PLT=1,HARDCOPY;
```

```
ERASE
RETURN
```

45 Degree Coordinated Turn Plots

```
//R8.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE 45 DEGREE TURN
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,UPPER');
PLOT(T1,Y(:,[10])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         YAW RATE (R)/,LOWER,KEEP')
IF PLT=1,HARDCOPY
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/          VELOCITIES (U) AND (W)/,...
    YMIN=-100,YMAX=2,UPPER LEFT')
PLOT(T1,Y(:,[8 12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        PITCH ANGLE VS PITCH RATE/,...
    YMIN=-2,YMAX=2,UPPER RIGHT,KEEP')
PLOT(T1,Y(:,13)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         ROLL RATE/,...
    YMAX=100,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,...
    YMAX=3,YMIN=-3,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT CANARD DEFLECTION/,UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT CANARD RATE/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RIGHT CANARD RATE/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
```

```
    YLABEL/DEG-SEC/,TITLE/        LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT TRAILING EDGE RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RUDDER DEFLECTION/,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        RUDDER RATE/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,19)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         BANK ANGLE COMMAND/,UPPER');
PLOT(T1,Y(:,20)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        YAW RATE COMMAND/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

Beta Command Plots

```
//R10.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE 45 DEGREE TURN
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,UPPER');
PLOT(T1,Y(:,[10])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         YAW RATE (R)/,LOWER,KEEP')
IF PLT=1,HARDCOPY
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/          VELOCITIES (U) AND (W)/,...
    YMIN=-100,YMAX=2,UPPER LEFT')
PLOT(T1,Y(:,[8 12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/       PITCH ANGLE VS PITCH RATE/,...
    YMIN=-2,YMAX=2,UPPER RIGHT,KEEP')
PLOT(T1,Y(:,13)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        ROLL RATE/,...
    YMAX=100,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/        NORMAL ACCELERATION/,...
    YMAX=3,YMIN=-3,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
```

```
    YLABEL/DEGREES/,TITLE/        LEFT CANARD DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT CANARD RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RIGHT CANARD RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT TRAILING EDGE RATE/,...
    YMAX=20,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RIGHT TRAILING EDGE RATE/,...
    YMAX=20,YMIN=-20,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         RUDDER DEFLECTION/,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RUDDER RATE/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,17)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/           YAW ANGLE COMMAND/,UPPER');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

F.4 Discrete Time System Analysis Macros

The discrete time analysis of the system with the PI control law and the sampling time equal to 40 Hz is simulated in MATRIX_x using the following macros.

```
//DISCRETE ANALYSIS WITH TSAMP=40 HZ
//THIS PROGRAM SIMULATES THE SYSTEM WITH ACTUATORS
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.025;
LGT1=(LGT-.5)/.025;//SETS THE RAMPING FOR SURFACE LIMITS
T1=<0:.025:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=.03491*<0:.025:3>';
    R2B=<R2A;3*.03491*ONES(LGT2,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
    R8A =2*.7854*<0:.025:.5>';
    R8A1=2*.02542*<0:.025:.5>';
    R8B =<R8A;.7854*ONES(LGT1,1)>;
    R8B1=<R8A1;.02542*ONES(LGT1,1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),R8B,R8B1>;
    R10A=2*.0349*<0:.025:.5>';
    R10B=<R10A;.0349*ONES(LGT1,1)>;
R10=<0*ONES(T1),R10B,0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B R8A R8A1 R8B R8B1 R10A R10B
//
G1=40;//SAMPLING FREQUENCY
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
INQUIRE L 'WHAT IS YOUR LAMBDA 0 MULTIPLIER: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
M(3,1)=KQ;M(4,2)=KP;
SIGMA=<SIG1,0,0,0,0;0,SIG2,0,0,0;0,0,SIG3,0,0;0,0,0,SIG4,0;...
       0,0,0,0,SIG5>;
TITLE1='THE FOLLOWING ARE THE PARAMETERS USED IN THE SIMULATION';
DISPLAY(TITLE1)
G=G1*EYE(5);
SIGMA
```

```
F2=C2+M*A12;
K1=INV(F2*B2)*SIGMA
K2=L*K1
EXEC('M12');//ACTUATOR BUILD
EXEC('M8');//PI CONTROLLER BUILD
EXEC('M7');//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6');//PLANT MATRIX WITH ACTUATORS
EXEC('M11');//MULTI SUPERBLOCK WITH ACTUATORS
EXEC('M19');//PRE FILTERS BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH PREFILTERS
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Main Program for CRCA Simulation - Model Following Input

```
//DISCRETE ANALYSIS WITH TSAMP=40 HZ
//MULTI1.DAT - SIMULATION WITH MODEL FOLLOWING
//SIMULATION WITH ACTUATORS
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.025;
T1=<0:.025:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=.03491*<0:.025:3>';
    R2B=<R2A;(3*.03491)*ONES(LGT2,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
R3 =<0*ONES(T1),0*ONES(T1),R3B,0*ONES(T1),0*ONES(T1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),.7854*ONES(T1),.02542*ONES(T1)>;
R10=<0*ONES(T1),.0349*ONES(T1),0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B
//
G1=40;//SAMPLING FREQUENCY GAIN
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
```

```
INQUIRE L 'WHAT IS YOUR LAMBDA O MULTIPLIER: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
M(3,1)=KQ;M(4,2)=KP;
SIGMA=<SIG1,0,0,0,0;0,SIG2,0,0,0;0,0,SIG3,0,0;0,0,0,SIG4,0;...
       0,0,0,0,SIG5>;
TITLE1='THE FOLLOWING ARE THE PARAMETERS USED IN THE SIMULATION';
DISPLAY(TITLE1)
G=G1*EYE(5);
SIGMA
F2=C2+M*A12;
K1=INV(F2*B2)*SIGMA
K2=L*K1
EXEC('M12');//ACUATOR BUILD
EXEC('M8') ;//PI CONTROLLER
EXEC('M7') ;//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6') ;//PLANT WITH ACTUATORS SUPER BLOCK
EXEC('M11');//MULTI SUPERBLOCK WITH ACTUATORS
EXEC('M15');//MODEL FOLLOWING BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH MODEL FOLLOWING
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Aircraft Model with Actuators

```
//M6.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A SYSTEM MATRIX SUPER BLOCK WITH ACTUATORS
//MAKE SURE THE S MATRIX IS DEFINED
//ADDS PLANT MATRIX
PLANT1
0
DEF,2, SUPER,   NAME,PLANT,         INPUTS,5, OUT,12,ENTRY
DEF,1, SUPER,   NAME,ACT,           INPUTS,5, OUT,10,ENTRY
CONNECT
INPUT,5,1,Y,    INTERNAL,1,2,2,1,4,2,6,3,8,4,10,5,0,0

OUTPUT,22,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUTPUT,22,1,1,13,2,14,3,15,4,16,5,17,6,18,7,19,8,20,9,21,10,22,0,0
TOP
```

```
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Aircraft Model without Actuators

```
//M7.DAT
BUILD
PLOT OFF
EDIT
PLANT
0
DEF,1, DYN,        STATE, NAME,B,    INPUTS,5,  OUT,8,STATES,0,ENTRY,B
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1, INPUTS,2,  OUT,8,ENTRY,2,1,1
DEF,3, DYN,        NTH,   NAME,INT,
OUT,8,ENTRY,1,1,1,1,1,1,1,1,1,Y
DEF,4, DYN,        STATE, NAME,C,    INPUTS,8,  OUT,10,STATES,0,ENTRY,C
DEF,5, DYN,        STATE, NAME,A,    INPUTS,8,  OUT,8,STATES,0,ENTRY,A
CONNECT
INPUT,5,1,Y,       INTERNAL,1,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,0,0
                   INTERNAL,2,3,Y
                   INTERNAL,3,4,Y
                   INTERNAL,3,5,Y
                   INTERNAL,5,2,1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,0,0
OUTPUT,12,4,1,1,2,2,3,3,4,4,5,5,6,8,7,9,8,10,9,11,10,12,0,0
OUTPUT,12,2,1,6,2,7,0,0
//OUTPUT FROM BLOCK 2 - 6 AND 7 ARE XDOT(A11 AND A12)
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

PI Controller

```
//M8.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A PI CONTROLLER
//THE CURRENT INPUTS ARE 5 WITH K1 AND K2
//DIMENSIONED 5 X 5
```

```
//TO CHANGE THEM FOR A 3 INPUT SYSTEM JUST
//CHANGE OUTPUTS AND INPUTS FROM 5 TO WHATEVER YOU WANT
PI
.025
.025
DEF,1, DYN, Nth,  NAME,INT,OUT,5,ENTRY,1,.025,.025,.025,.025,.025,YES
DEF,2, DYN,       STATE,NAME,K2, INPUTS,5,OUT,5,STATES,0,ENTRY,K2
DEF,5, DYN,       STATE,NAME,K1, INPUTS,5,OUT,5,STATES,0,ENTRY,K1
DEF,3, ALGEBRAIC, SUM,  NAME,SUM1,INPUTS,2,OUT,5,ENTRY,2,1,1
CONNECT
INPUT,5,1,Y,             INTERNAL,1,2,Y,          OUTPUT,5,3,Y
INPUT,5,5,Y,             INTERNAL,2,3,1,1,2,2,3,3,4,4,5,5,0,0
                         INTERNAL,5,3,1,6,2,7,3,8,4,9,5,10,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Plant and PI Controller

```
//M11.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A MULTI SUPERBLOCK FOR SIMULATION - 5 INPUTS
//INSURE YOU HAVE CREATED THE SUPERBLOCKS PI AND APPROPRIATE PLANT
//THIS IS A SIMULATION FOR AN IRREGULAR PLANT WITH ACTUATORS
//THIS SUPERBLOCK WILL HAVE 10 OUTPUTS- OUTPUT 1-5 WILL BE COMMAND INPUTS
//OUTPUTS 5-14 WILL BE STATE VARIABLES
MULTI
0
DEF,2, SUPER,           NAME,PI,    INPUTS,5,       OUT,5,          ENTRY
DEF,4, SUPER,           NAME,PLANT1,INPUTS,5,       OUT,22,         ENTRY
DEF,3, DYN,       STATE,NAME,G,     INPUTS,5,       OUT,5,STATES,0,ENTRY,G
DEF,5, DYN,       STATE,NAME,M,     INPUTS,2,       OUT,5,STATES,0,ENTRY,M
DEF,1, ALGEBRAIC, SUM,  NAME,SUM1,  INPUTS,2,       OUT,5,ENTRY,2,1,-1
DEF,6, ALGEBRAIC, SUM,  NAME,W,     INPUTS,2,       OUT,5,ENTRY,2,1,1
CONNECT
INPUT,5,1,1,1,2,2,3,3,4,4,5,5,0,0,
                              INTERNAL,1,2,Y
                              INTERNAL,2,3,Y
                              INTERNAL,3,4,Y
                              INTERNAL,4,5,6,1,7,2,0,0
```

```
                                  INTERNAL,4,6,1,1,2,2,3,3,4,4,5,5,0,0
                                  INTERNAL,5,6,1,6,2,7,3,8,4,9,5,10,0,0
                                  INTERNAL,6,1,1,6,2,7,3,8,4,9,5,10,0,0
OUTPUT,20,4,14,1,16,2,18,3,20,4,22,5,0,0
OUTPUT,20,4,13,16,15,17,17,18,19,19,21,20,0,0
OUTPUT,20,4,1,6,2,7,3,8,4,9,5,10,8,11,9,12,10,13,11,14,12,15,0,0

//OUTPUTS 1-5 ARE CONTROL OUTPUTS:CL,CR,TEL,TER,RUDDER
//OUTPUTS 6-15 ARE STATE OUTPUTS:VEL,THETA,BETA,PHI,R,W,Q,P,NZ,NZPILOT
//OUTPUTS 13,15,17,19,21 RATES FOR ACTUATORS CL,CR,TEL,TER,RUD
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Actuators

```
//M12.DAT
//THIS ROUTINE BUILDS THE ACTUATORS FOR THE CRCA
//FIRST ORDER MODEL OF XD=-20X+20U
BACT=20;//B MATRIX FOR THE ACTUATOR
AACT=-20;//A MATRIX FOR THE ACTUATOR
BUILD
PLOT OFF
EDIT
ACL
O
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,  OUT,1,ENTRY,30,-60,1,Y
DEF,5, DYN,       STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                   INTERNAL,1,2,1,1,0,0
                             INTERNAL,2,3
                             INTERNAL,3,4
                             INTERNAL,4,5
                             INTERNAL,5,2,1,2,0,0
OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP
```

```
//BUILDING THE RIGHT CANARD ACTUATOR
COPY,       ACL,       ACR

//BUILDING TRAILING EDGE LEFT ACTUATOR
EDIT
ATL1
0
DEF,1, DYN,        STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,      3  ,  NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,   LIM,   NAME,LMTINT,INPUTS,1,  OUT,1,ENTRY,30,-30,1,Y
DEF,5, DYN,        STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                INTERNAL,1,2,1,1,0,0
                          INTERNAL,2,3
                          INTERNAL,3,4
                          INTERNAL,4,5
                          INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING TRAILING EDGE RIGHT ACTUATOR
COPY,        ATL1,      ATR1

//BUILDING RUDDER ACTUATOR
EDIT
ATRUD
0
DEF,1, DYN,        STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,      3  ,  NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,   LIM,   NAME,LMTINT,INPUTS,1,  OUT,1,ENTRY,20,-20,1,Y
DEF,5, DYN,        STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                INTERNAL,1,2,1,1,0,0
                          INTERNAL,2,3
                          INTERNAL,3,4
                          INTERNAL,4,5
                          INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
```

```
TOP

//BUILD SUPER BLOCK OF ACTUATORS
EDIT
ACT
O
DEF, 1, SUPER,          NAME,ACL,   INPUTS,1, OUT,2,ENTRY
DEF, 2, SUPER,          NAME,ACR,   INPUTS,1, OUT,2,ENTRY
DEF, 3, SUPER,          NAME,ATL1,  INPUTS,1, OUT,2,ENTRY
DEF, 4, SUPER,          NAME,ATR1,  INPUTS,1, OUT,2,ENTRY
DEF, 5, SUPER,          NAME,ATRUD, INPUTS,1, OUT,2,ENTRY
CONNECT
INPUT,5,1,1,1,0,0,                  OUTPUT,10,1,1,1,2,2,0,0
INPUT,5,2,2,1,0,0,                  OUTPUT,10,2,1,3,2,4,0,0
INPUT,5,3,3,1,0,0,                  OUTPUT,10,3,1,5,2,6,0,0
INPUT,5,4,4,1,0,0,                  OUTPUT,10,4,1,7,2,8,0,0
INPUT,5,5,5,1,0,0,                  OUTPUT,10,5,1,9,2,10,0,0

TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR BACT AACT
```

Model Following Filter

```
//M15.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A MODEL FOLLOWING PROGRAM
PRFB=<-12,-57,-100,1;1,0,0,0;0,1,0,0;0,0,100,0>;
PRFR=PRFB;
PRFP=<-18,-105,-250,1;1,0,0,0;0,1,0,0;0,0,250,0>;
PRFV=PRFP;
PRFT=<-10,-41,-50,1;1,0,0,0;0,1,0,0;0,0,50,0>;
BUILD
PLOT OFF
EDIT
PREF
O
DEF,1, DYN,  STATE,NAME,  PRE1,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFV,Y
DEF,2, DYN,  STATE,NAME,  PRE2,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFB,Y
DEF,3, DYN,  STATE,NAME,  PRE3,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFT,Y
DEF,4, DYN,  STATE,NAME,  PRE4,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFP,Y
DEF,5, DYN,  STATE,NAME,  PRE5,  INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFR,Y
```

```
CONNECT
INPUT,5,1,1,1,0,0,                    OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                    OUT,5,2 1,2,0,0
INPUT,5,3,3,1,0,0,                    OUT,5,3,1,3,0,0
INPUT,5,4,4.1,0,0,                    OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                    OUT,5,5,1,5,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR PRFV PRFB PRFT PRFP PRFR
```

Aircraft Model with PI Controller and Pre-Filter

```
//M16.DAT
//THIS EXECUTABLE FILE BUILDS A MULTI SUPERBLOCK WITH PRE FILTER
//ALSO BUILDS NORMAL ACCELERATION AT PILOT STATION OUTPUT
BUILD
PLOT OFF
EDIT
MULTI1
0
DEF,2,  SUPER,      NAME,MULTI,      INPUTS,5,  OUT,20,  ENTRY
DEF,1,  SUPER,      NAME,PREF,       INPUTS,5,  OUT,5,   ENTRY
DEF,5,  DYN,STATE,  NAME,NZM,        INPUTS,4,  OUT,1,STATES,0,ENTRY,NZM
DEF,4,  ALGEBRAIC,  SUM,NAME,NZ,     INPUTS,2,  OUT,1,ENTRY,2,1,1
CONNECT
INPUT,5,1,Y,      INTERNAL,1,2,Y
                  INTERNAL,2,5,1,1,2,2,3,3,4,4,0,0
                  INTERNAL,5,4,1,1,0,0
                  INTERNAL,2,4,14,2,0,0
OUT,25,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUT,25,2,13,13,14,14,0,0
OUT,25,4,1,15,0,0
OUT,25,1,1,16,2,17,3,18,4,19,5,20,0,0
OUT,25,2,16,21,17,22,18,23,19,24,20,25,0,0
//OUTPUTS 21,22,23,24,25 ARE RATES FOR CL,CR,TEL,TER,RUD
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Simulation and Plotting

```
//M18.DAT
ERASE;CLEAR Y
INQUIRE TYPE 'WHAT FLIGHT MANEUVER WOULD YOU LIKE: '
Y=HSIM(T1,TYPE);
INQUIRE PLT 'DO YOU WANT HARDCOPIES'
INQUIRE PLT1 '1=R8 PLOTS----- 0=R2 PLOTS----- 2=R10 PLOTS';
IF PLT1=1,EXEC('R8')
IF PLT1=0,EXEC('R3')
IF PLT1=2,EXEC('R10')
INQUIRE REST 'DO YOU WANT TO CHANGE THE INPUT: '
IF REST=YES,EXEC('M18');END;
```

Low Pass Filter

```
//M19.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A LOW PASS PREFILTER
PRF=<-10,1;10,0>;
BUILD
PLOT OFF
EDIT
PREF
0
DEF,1, DYN,  STATE,NAME,  PRE1,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,2, DYN,  STATE,NAME,  PRE2,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,3, DYN,  STATE,NAME,  PRE3,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,4, DYN,  STATE,NAME,  PRE4,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,5, DYN,  STATE,NAME,  PRE5,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
CONNECT
INPUT,5,1,1,1,0,0,                           OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                           OUT,5,2,1,2,0,0
INPUT,5,3,3,1,0,0,                           OUT,5,3,1,3,0,0
INPUT,5,4,4,1,0,0,                           OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                           OUT,5,5,1,5,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR PRF
```

Pitch Rate Tracking Plots

```
//R2.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE PITCH RATE TRACKING
PLOT(T1,Y(:,[8])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          PITCH ANGLE/,UPPER')
PLOT(T1,Y(:,[12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          PITCH RATE/,LOWER,KEEP')
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,...
    YMIN=-1,YMAX=1,UPPER LEFT');
PLOT(T1,Y(:,[10 13])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        ROLL RATE (P) VS YAW RATE (R)/,...
    YMIN=-1,YMAX=1,LOWER LEFT,KEEP')
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/         VELOCITIES (U) AND (W)/,...
    UPPER RIGHT')
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       LEFT CANARD DEFLECTION/,UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT CANARD RATE/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT CANARD RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT TRAILING EDGE RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
```

```
    YLABEL/DEGREES/,TITLE/          RUDDER DEFLECTION/,...
    YMIN=-1,YMAX=1,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RUDDER RATE/,...
    YMIN=-1,YMAX=1,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,18)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          PITCH RATE COMMAND/,UPPER');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

45 Degree Coordinated Turn Plots

```
//R8.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE 45 DEGREE TURN
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         PHI VS BETA/,UPPER');
PLOT(T1,Y(:,[10])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          YAW RATE (R)/,LOWER,KEEP')
IF PLT=1,HARDCOPY
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/           VELOCITIES (U) AND (W)/,...
    YMIN=-100,YMAX=2,UPPER LEFT')
PLOT(T1,Y(:,[8 12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        PITCH ANGLE VS PITCH RATE/,...
    YMIN=-2,YMAX=2,UPPER RIGHT,KEEP')
PLOT(T1,Y(:,13)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         ROLL RATE/,...
    YMAX=100,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,...
    YMAX=3,YMIN=-3,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT CANARD DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT CANARD RATE/,...
```

```
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT CANARD RATE/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RIGHT TRAILING EDGE RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         RUDDER DEFLECTION/,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RUDDER RATE/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,19)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          BANK ANGLE COMMAND/,UPPER');
PLOT(T1,Y(:,20)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         YAW RATE COMMAND/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

Beta Command Plots

```
//R10.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE 45 DEGREE TURN
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,UPPER');
PLOT(T1,Y(:,[10])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          YAW RATE (R)/,LOWER,KEEP')
IF PLT=1,HARDCOPY
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/           VELOCITIES (U) AND (W)/,...
    YMIN=-100,YMAX=2,UPPER LEFT')
```

```
PLOT(T1,Y(:,[8 12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        PITCH ANGLE VS PITCH RATE/,...
    YMIN=-2,YMAX=2,UPPER RIGHT,KEEP')
PLOT(T1,Y(:,13)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         ROLL RATE/,...
    YMAX=100,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,...
    YMAX=3,YMIN=-3,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT CANARD DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT CANARD RATE/,...
    YMAX=50,YMIN=-50,LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RIGHT CANARD RATE/,...
    YMAX=50,YMIN=-50,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         LEFT TRAILING EDGE RATE/,...
    YMAX=50,YMIN=-50,LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RIGHT TRAILING EDGE RATE/,...
    YMAX=50,YMIN=-50,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         RUDDER DEFLECTION/,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RUDDER RATE/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,17)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/           YAW ANGLE COMMAND/,UPPER');
IF PLT=1,HARDCOPY;
ERASE
```

RETURN

F.5 ARMA Fixed Gain System Analysis Macros

The ARMA plant representations were analyzed in the discrete time domain with the fixed gain PI control law and the sampling time equal to 40 Hz with the following $MATRIX_x$ macros.

```
//DISCRETE ANALYSIS WITH TSAMP=.025 AND ARMA MODEL REPRESENTATION
EXEC('ARMA')
CLEAR CHECK CO TT R SD AD BD CD DD V1 VI TF T
CLEAR RN CBAR ADBAR BDBAR H SQ SP1 SP2 GON GODEN
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.025;
LGT1=(LGT-.5)/.025;//SETS THE RAMPING FOR SURFACE LIMITS
T1=<0:.025:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=.03491*<0:.025:3>';
    R2B=<R2A;(3*.03491)*ONES(LGT2,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
    R8A =2*.7854*<0:.025:.5>';
    R8A1=2*.02542*<0:.025:.5>';
    R8B =<R8A;.7854*ONES(LGT1,1)>;
    R8B1=<R8A1;.02542*ONES(LGT1,1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),R8B,R8B1>;
    R10A=2*.0349*<0:.025:.5>';
    R10B=<R10A;.0349*ONES(LGT1,1)>;
R10=<0*ONES(T1),R10B,0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B R8A R8A1 R8B R8B1 R10A R10B
//
G1=1;//FIXED AS A FUNCTION OF SAMPLING TIME
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
INQUIRE GAM1 'WHAT IS YOUR GAMMA-1 VALUE: '
INQUIRE GAM2 'WHAT IS YOUR GAMMA-2 VALUE: '
INQUIRE GAM3 'WHAT IS YOUR GAMMA-3 VALUE: '
INQUIRE GAM4 'WHAT IS YOUR GAMMA-4 VALUE: '
INQUIRE GAM5 'WHAT IS YOUR GAMMA-5 VALUE: '
M(3,1)=KQ;M(4,2)=KP;
```

```
GAMMA=<GAM1,0,0,0,0;0,GAM2,0,0,0;0,0,GAM3,0,0;0,0,0,GAM4,0;...
       0,0,0,0,GAM5>;
SIGMA=<SIG1,0,0,0,0;0,SIG2,0,0,0;0,0,SIG3,0,0;0,0,0,SIG4,0;...
       0,0,0,0,SIG5>;
CLEAR SIG1 SIG2 SIG3 SIG4 SIG5 GAM1 GAM2 GAM3 GAM4 GAM5
G=G1*EYE(5);
K1=INV(B1ARMA)*SIGMA;
K2=INV(GO)*GAMMA;
EXEC('M12');//ACTUATOR BUILD
EXEC('M8') ;//PI CONTROLLER
EXEC('M7') ;//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6') ;//PLANT WITH ACTUATORS SUPER BLOCK
EXEC('M11');//MULTI SUPERBLOCK W/O ACTUATORS
EXEC('M19');//PRE FILTERS BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH PREFILTERS
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Main Program for CRCA Simulation - Model Following Input

```
//DISCRETE ANALYSIS WITH TSAMP=.025 WITH ARMA GAIN
EXEC('ARMA')
CLEAR CHECK CO TT R SD AD BD CD DD V1 VI TF T
CLEAR RN CBAR ADBAR BDBAR H SQ SP1 SP2 GON GODEN
//MULTI1.DAT - SIMULATION WITH MODEL FOLLOWING
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.025;
T1=<0:.025:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=.03491*<0:.025:3.0>';
    R2B=<R2A;(3*.03491)*ONES(LGT2,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),.7854*ONES(T1),.02542*ONES(T1)>;
R10=<0*ONES(T1),.0349*ONES(T1),0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B
//
```

```
G1=1;//FIXED AT THE SAMPLING FREQUENCY
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
INQUIRE GAM1 'WHAT IS YOUR GAMMA-1 VALUE: '
INQUIRE GAM2 'WHAT IS YOUR GAMMA-2 VALUE: '
INQUIRE GAM3 'WHAT IS YOUR GAMMA-3 VALUE: '
INQUIRE GAM4 'WHAT IS YOUR GAMMA-4 VALUE: '
INQUIRE GAM5 'WHAT IS YOUR GAMMA-5 VALUE: '
M(3,1)=KQ;M(4,2)=KP;
GAMMA=<GAM1,0,0,0,0;0,GAM2,0,0,0;0,0,GAM3,0,0;0,0,0,GAM4,0;...
       0,0,0,0,GAM5>;
SIGMA=<SIG1,0,0,0,0;0,SIG2,0,0,0;0,0,SIG3,0,0;0,0,0,SIG4,0;...
       0,0,0,0,SIG5>;
CLEAR GAM1 GAM2 GAM3 GAM4 GAM5 SIG1 SIG2 SIG3 SIG4 SIG5
TITLE1='THE FOLLOWING ARE THE PARAMETERS USED IN THE SIMULATION';
DISPLAY(TITLE1)
G=G1*EYE(5);
SIGMA
GAMMA
K1=INV(B1ARMA) *SIGMA
K2=INV(GO)*GAMMA
EXEC('M12');//ACTUATOR BUILD
EXEC('M8') ;//PI CONTROLLER
EXEC('M7') ;//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6') ;//PLANT WITH ACTUATORS SUPER BLOCK
EXEC('M11');//MULTI SUPERBLOCK W/O ACTUATORS
EXEC('M15');//PRE FILTERS BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH MODEL FOLLOWING
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Aircraft Model With Actuators

```
//M6.DAT
BUILD
PLOT OFF
EDIT
```

```
//THE FOLLOWING CREATES A SYSTEM MATRIX SUPER BLOCK WITH ACTUATORS
//MAKE SURE THE S MATRIX IS DEFINED
//ADDS PLANT MATRIX
PLANT1
0
DEF,2, SUPER,   NAME,PLANT,         INPUTS,5, OUT,12,ENTRY
DEF,1, SUPER,   NAME,ACT,           INPUTS,5, OUT,10,ENTRY
CONNECT
INPUT,5,1,Y,    INTERNAL,1,2,2,1,4,2,6,3,8,4,10,5,0,0

OUTPUT,22,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUTPUT,22,1,1,13,2,14,3,15,4,16,5,17,6,18,7,19,8,20,9,21,10,22,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Aircraft Model Without Actuators

```
//M7.DAT
BUILD
PLOT OFF
EDIT
PLANT
0
DEF,1, DYN,       STATE, NAME,B,    INPUTS,5,  OUT,8,STATES,0,ENTRY,B
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1, INPUTS,2,  OUT,8,ENTRY,2,1,1
DEF,3, DYN,       NTH,   NAME,INT,
OUT,8,ENTRY,1,1,1,1,1,1,1,1,1,Y
DEF,4, DYN,       STATE, NAME,C,    INPUTS,8,  OUT,10,STATES,0,ENTRY,C
DEF,5, DYN,       STATE, NAME,A,    INPUTS,8,  OUT,8,STATES,0,ENTRY,A
CONNECT
INPUT,5,1,Y,      INTERNAL,1,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,0,0
                  INTERNAL,2,3,Y
                  INTERNAL,3,4,Y
                  INTERNAL,3,5,Y
                  INTERNAL,5,2,1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,0,0
OUTPUT,12,4,1,1,2,2,3,3,4,4,5,5,6,8,7,9,8,10,9,11,10,12,0,0
OUTPUT,12,2,1,6,2,7,0,0
//OUTPUT FROM BLOCK 2 - 6 AND 7 ARE XDOT(A11 AND A12)
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

PI Controller

```
//M8.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A PI CONTROLLER
//THE CURRENT INPUTS ARE 5 WITH K1 AND K2
//DIMENSIONED 5 X 5
//TO CHANGE THEM FOR A 3 INPUT SYSTEM JUST
//CHANGE OUTPUTS AND INPUTS FROM 5 TO WHATEVER YOU WANT
PI
.025
.025
DEF,1, DYN,Nth,   NAME,INT,OUT,5,ENTRY,1,.025,.025,.025,.025,.025,YES
DEF,2, DYN,       STATE,NAME,K2,  INPUTS,5,OUT,5,STATES,0,ENTRY,K2
DEF,5, DYN,       STATE,NAME,K1,  INPUTS,5,OUT,5,STATES,0,ENTRY,K1
DEF,3, ALGEBRAIC, SUM,  NAME,SUM1, INPUTS,2,OUT,5,ENTRY,2,1,1
CONNECT
INPUT,5,1,Y,              INTERNAL,1,2,Y,           OUTPUT,5,3,Y
INPUT,5,5,Y,              INTERNAL,2,3,1,1,2,2,3,3,4,4,5,5,0,0
                          INTERNAL,5,3,1,6,2,7,3,8,4,9,5,10,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Plant and PI Controller

```
//M11.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A MULTI SUPERBLOCK FOR SIMULATION - 5 INPUTS
//INSURE YOU HAVE CREATED THE SUPERBLOCKS PI AND APPROPRIATE PLANT
//THIS IS A SIMULATION FOR AN IRREGULAR PLANT WITH ACTUATORS
//THIS SUPERBLOCK WILL HAVE 10 OUTPUTS- OUTPUT 1-5 WILL BE COMMAND INPUTS
//OUTPUTS 5-14 WILL BE STATE VARIABLES
MULTI
0
DEF,2, SUPER,          NAME,PI,     INPUTS,5,       OUT,5,          ENTRY
DEF,4, SUPER,          NAME,PLANT1,INPUTS,5,        OUT,22,         ENTRY
```

```
DEF,3, DYN,        STATE,NAME,G,     INPUTS,5,       OUT,5,STATES,0,ENTRY,G
DEF,5, DYN,        STATE,NAME,M,     INPUTS,2,       OUT,5,STATES,0,ENTRY,M
DEF,1, ALGEBRAIC, SUM,  NAME,SUM1,  INPUTS,2,       OUT,5,ENTRY,2,1,-1
DEF,6, ALGEBRAIC, SUM,  NAME,W,     INPUTS,2,       OUT,5,ENTRY,2,1,1
CONNECT
INPUT,5,1,1,1,2,2,3,3,4,4,5,5,0,0,
                             INTERNAL,1,2,Y
                             INTERNAL,2,3,Y
                             INTERNAL,3,4,Y
                             INTERNAL,4,5,6,1,7,2,0,0
                             INTERNAL,4,6,1,1,2,2,3,3,4,4,5,5,0,0
                             INTERNAL,5,6,1,6,2,7,3,8,4,9,5,10,0,0
                             INTERNAL,6,1,1,6,2,7,3,8,4,9,5,10,0,0
OUTPUT,20,4,14,1,16,2,18,3,20,4,22,5,0,0
OUTPUT,20,4,13,16,15,17,17,18,19,19,21,20,0,0
OUTPUT,20,4,1,6,2,7,3,8,4,9,5,10,8,11,9,12,10,13,11,14,12,15,0,0

//OUTPUTS 1-5 ARE CONTROL OUTPUTS:CL,CR,TEL,TER,RUDDER
//OUTPUTS 6-15 ARE STATE OUTPUTS:VEL,THETA,BETA,PHI,R,W,Q,P,NZ,NZPILOT
//OUTPUTS 13,15,17,19,21 RATES FOR ACTUATORS CL,CR,TEL,TER,RUD
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Actuators

```
//M12.DAT
//THIS ROUTINE BUILDS THE ACTUATORS FOR THE CRCA
//FIRST ORDER MODEL OF XD=-20X+20U
BACT=20;//B MATRIX FOR THE ACTUATOR
AACT=-20;//A MATRIX FOR THE ACTUATOR
BUILD
PLOT OFF
EDIT
ACL
0
DEF,1, DYN,        STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,    NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,      3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,   LIM,   NAME,LMTINT,INPUTS,1,  OUT,1,ENTRY,30,-60,1,Y
DEF,5, DYN,        STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
```

```
INPUT,1,1,                INTERNAL,1,2,1,1,0,0
                          INTERNAL,2,3
                          INTERNAL,3,4
                          INTERNAL,4,5
                          INTERNAL,5,2,1,2,0,0
OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING THE RIGHT CANARD ACTUATOR
COPY,       ACL,        ACR

//BUILDING TRAILING EDGE LEFT ACTUATOR
EDIT
ATL1
0
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,   OUT,1,ENTRY,30,-30,1,Y
DEF,5, DYN,       STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                INTERNAL,1,2,1,1,0,0
                          INTERNAL,2,3
                          INTERNAL,3,4
                          INTERNAL,4,5
                          INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING TRAILING EDGE RIGHT ACTUATOR
COPY,        ATL1,      ATR1

//BUILDING RUDDER ACTUATOR
EDIT
ATRUD
0
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,   OUT,1,ENTRY,20,-20,1,Y
```

```
DEF,5, DYN,       STATE, NAME,A,     INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                INTERNAL,1,2,1,1,0,0
                          INTERNAL,2,3
                          INTERNAL,3,4
                          INTERNAL,4,5
                          INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILD SUPER BLOCK OF ACTUATORS
EDIT
ACT
0
DEF, 1, SUPER,            NAME,ACL,   INPUTS,1, OUT,2,ENTRY
DEF, 2, SUPER,            NAME,ACR,   INPUTS,1, OUT,2,ENTRY
DEF, 3, SUPER,            NAME,ATL1,  INPUTS,1, OUT,2,ENTRY
DEF, 4, SUPER,            NAME,ATR1,  INPUTS,1, OUT,2,ENTRY
DEF, 5, SUPER,            NAME,ATRUD, INPUTS,1, OUT,2,ENTRY
CONNECT
INPUT,5,1,1,1,0,0,                    OUTPUT,10,1,1,1,2,2,0,0
INPUT,5,2,2,1,0,0,                    OUTPUT,10,2,1,3,2,4,0,0
INPUT,5,3,3,1,0,0,                    OUTPUT,10,3,1,5,2,6,0,0
INPUT,5,4,4,1,0,0,                    OUTPUT,10,4,1,7,2,8,0,0
INPUT,5,5,5,1,0,0,                    OUTPUT,10,5,1,9,2,10,0,0

TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR BACT AACT
```

Model Following Filter

```
//M15.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A MODEL FOLLOWING PROGRAM
PRFB=<-12,-57,-100,1;1,0,0,0;0,1,0,0;0,0,100,0>;
PRFR=PRFB;
PRFP=<-18,-105,-250,1;1,0,0,0;0,1,0,0;0,0,250,0>;
PRFV=PRFP;
PRFT=<-10,-41,-50,1;1,0,0,0;0,1,0,0;0,0,50,0>;
```

```
BUILD
PLOT OFF
EDIT
PREF
0
DEF,1, DYN,  STATE,NAME,  PRE1,   INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFV,Y
DEF,2, DYN,  STATE,NAME,  PRE2,   INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFB,Y
DEF,3, DYN,  STATE,NAME,  PRE3,   INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFT,Y
DEF,4, DYN,  STATE,NAME,  PRE4,   INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFP,Y
DEF,5, DYN,  STATE,NAME,  PRE5,   INPUTS,1,  OUT,1, STATES,3,ENTRY,PRFR,Y
CONNECT
INPUT,5,1,1,1,0,0,                           OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                           OUT,5,2,1,2,0,0
INPUT,5,3,3,1,0,0,                           OUT,5,3,1,3,0,0
INPUT,5,4,4,1,0,0,                           OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                           OUT,5,5,1,5,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR PRFV PRFB PRFT PRFP PRFR
```

Aircraft Model With PI Controller and Pre-Filter

```
//M16.DAT
//THIS EXECUTABLE FILE BUILDS A MULTI SUPERBLOCK WITH PRE FILTER
//ALSO BUILDS NORMAL ACCELERATION AT PILOT STATION OUTPUT
BUILD
PLOT OFF
EDIT
MULTI1
0
DEF,2,  SUPER,      NAME,MULTI,        INPUTS,5,   OUT,20,  ENTRY
DEF,1,  SUPER,      NAME,PREF,         INPUTS,5,   OUT,5,   ENTRY
DEF,5,  DYN,STATE,  NAME,NZM,          INPUTS,4,   OUT,1,STATES,0,ENTRY,NZM
DEF,4,  ALGEBRAIC,  SUM,NAME,NZ,       INPUTS,2,   OUT,1,ENTRY,2,1,1
CONNECT
INPUT,5,1,Y,        INTERNAL,1,2,Y
                    INTERNAL,2,5,1,1,2,2,3,3,4,4,0,0
                    INTERNAL,5,4,1,1,0,0
                    INTERNAL,2,4,14,2,0,0
OUT,25,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUT,25,2,13,13,14,14,0,0
```

```
OUT,25,4,1,15,0,0
OUT,25,1,1,16,2,17,3,18,4,19,5,20,0,0
OUT,25,2,16,21,17,22,18,23,19,24,20,25,0,0
//OUTPUTS 21,22,23,24,25 ARE RATES FOR CL,CR,TEL,TER,RUD
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Simulation and Plotting

```
//M18.DAT
ERASE;CLEAR Y
INQUIRE TYPE 'WHAT FLIGHT MANEUVER WOULD YOU LIKE: '
SIM('IALG')
STIFF
Y=HSIM(T1,TYPE);
INQUIRE PLT 'DO YOU WANT HARDCOPIES'
INQUIRE PLT1 '1=R8 PLOTS----- 0=R2 PLOTS ---2=R10 PLOTS';
IF PLT1=1,EXEC('R8')
IF PLT1=0,EXEC('R2')
IF PLT1=2,EXEC('R10')
INQUIRE REST 'DO YOU WANT TO CHANGE THE INPUT: '
IF REST=YES,EXEC('M18');END;
```

Low-Pass Filtering

```
//M19.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A LOW PASS PREFILTER
PRF=<-10,1;10,0>;
BUILD
PLOT OFF
EDIT
PREF
0
DEF,1, DYN,  STATE,NAME,  PRE1,   INPUTS,1,  OUT,1,  STATES,1,ENTRY,PRF,Y
DEF,2, DYN,  STATE,NAME,  PRE2,   INPUTS,1,  OUT,1,  STATES,1,ENTRY,PRF,Y
DEF,3, DYN,  STATE,NAME,  PRE3,   INPUTS,1,  OUT,1,  STATES,1,ENTRY,PRF,Y
DEF,4, DYN,  STATE,NAME,  PRE4,   INPUTS,1,  OUT,1,  STATES,1,ENTRY,PRF,Y
DEF,5, DYN,  STATE,NAME,  PRE5,   INPUTS,1,  OUT,1,  STATES,1,ENTRY,PRF,Y
```

```
CONNECT
INPUT,5,1,1,1,0,0,                             OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                             OUT,5,2,1,2,0,0
INPUT,5,3,3,1,0,0,                             OUT,5,3,1,3,0,0
INPUT,5,4,4,1,0,0,                             OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                             OUT,5,5,1,5,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR PRF
```

Pitch Rate Tracking Plots

```
//R2.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE PITCH RATE TRACKING
PLOT(T1,Y(:,[8])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          PITCH ANGLE/,UPPER')
PLOT(T1,Y(:,[12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          PITCH RATE/,LOWER,KEEP')
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,...
    YMIN=-1,YMAX=1,UPPER LEFT');
PLOT(T1,Y(:,[10 13])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        ROLL RATE (P) VS YAW RATE (R)/,...
    YMIN=-1,YMAX=1,LOWER LEFT,KEEP')
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/         VELOCITIES (U) AND (W)/,...
    UPPER RIGHT')
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       LEFT CANARD DEFLECTION/,UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT CANARD RATE/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT CANARD RATE/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
```

```
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT TRAILING EDGE RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          RUDDER DEFLECTION/,...
    YMIN=-1,YMAX=1,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          RUDDER RATE/,...
    YMIN=-1,YMAX=1,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,18)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          PITCH RATE COMMAND/,UPPER');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

45 Degree Coordinated Turn Plots

```
//R8.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE 45 DEGREE TURN
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         PHI VS BETA/,UPPER');
PLOT(T1,Y(:,[10])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/          YAW RATE (R)/,LOWER,KEEP')
IF PLT=1,HARDCOPY
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/           VELOCITIES (U) AND (W)/,...
    YMIN=-100,YMAX=2,UPPER LEFT')
PLOT(T1,Y(:,[8 12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        PITCH ANGLE VS PITCH RATE/,...
    YMIN=-2,YMAX=2,UPPER RIGHT,KEEP')
PLOT(T1,Y(:,13)*57.3,'XLABEL/TIME IN SECONDS/,...
```

```
    YLABEL/DEG-SEC/,TITLE/        ROLL RATE/,...
    YMAX=100,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/        NORMAL ACCELERATION/,...
    YMAX=3,YMIN=-3,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT CANARD DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT CANARD RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT CANARD RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,...
    TITLE/            RIGHT TRAILING EDGE RATE/,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RUDDER DEFLECTION/,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        RUDDER RATE/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,19)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/         BANK ANGLE COMMAND/,UPPER');
PLOT(T1,Y(:,20)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        YAW RATE COMMAND/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

Beta Command Plots

```
//R10.DAT
//THIS ROUTINE PREPARES ALL PLOTS FOR THE 45 DEGREE TURN
PLOT(T1,Y(:,[9 7])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        PHI VS BETA/,UPPER');
PLOT(T1,Y(:,[10])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         YAW RATE (R)/,LOWER,KEEP')
IF PLT=1,HARDCOPY
PLOT(T1,Y(:,[6 11]),'XLABEL/TIME IN SECONDS/,...
    YLABEL/FT-SEC/,TITLE/          VELOCITIES (U) AND (W)/,...
    YMIN=-100,YMAX=2,UPPER LEFT')
PLOT(T1,Y(:,[8 12])*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/       PITCH ANGLE VS PITCH RATE/,...
    YMIN=-2,YMAX=2,UPPER RIGHT,KEEP')
PLOT(T1,Y(:,13)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        ROLL RATE/,...
    YMAX=100,YMIN=-20,LOWER LEFT,KEEP');
PLOT(T1,Y(:,15),'XLABEL/TIME IN SECONDS/,...
    YLABEL/G/,TITLE/         NORMAL ACCELERATION/,...
    YMAX=3,YMIN=-3,LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       LEFT CANARD DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       RIGHT CANARD DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,21),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT CANARD RATE/,...
    LOWER LEFT,KEEP');
PLOT(T1,Y(:,22),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT CANARD RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/       LEFT TRAILING EDGE DEFLECTION/,...
    UPPER LEFT');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/      RIGHT TRAILING EDGE DEFLECTION/,...
    UPPER RIGHT,KEEP');
PLOT(T1,Y(:,23),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        LEFT TRAILING EDGE RATE/,...
    LOWER LEFT,KEEP');
```

```
PLOT(T1,Y(:,24),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/         RIGHT TRAILING EDGE RATE/,...
    LOWER RIGHT,KEEP');
IF PLT=1,HARDCOPY;
PLOT (T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/        RUDDER DEFLECTION/,UPPER');
PLOT(T1,Y(:,25),'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEG-SEC/,TITLE/        RUDDER RATE/,LOWER,KEEP');
IF PLT=1,HARDCOPY;
PLOT(T1,Y(:,17)*57.3,'XLABEL/TIME IN SECONDS/,...
    YLABEL/DEGREES/,TITLE/          YAW ANGLE COMMAND/,UPPER');
IF PLT=1,HARDCOPY;
ERASE
RETURN
```

ARMA Calculation Macro

```
//THE FOLLOWING PROGRAM CALCULATES THE ARMA MODEL FOR THE PLANTS
CLEAR CO TT R SD
CDUM=<C1 C2>;
SDUM=<A,B;CDUM,0*EYE(5)>;
SD=DISC(SDUM,8,.025);
[AD BD CD DD]=SPLIT(SD,8);
CO=<CDUM*INV(AD)>;FOR I=2:8,CO=<CO;CDUM*INV(AD)**I>;END;
V1=0;CR=0;TT=<CO(1,:)>;R=TT;FOR I=2:16,R=<TT;CO(I,:)>;...
RN=RANK(R);IF RN=I-CR THEN TT=R;ELSEIF RN<I-CR THEN R=TT;...
VI=<V1;I>;V1=VI;CR=CR+1;END,END,T=R;CO;
TF=<T(1,:);T(6,:);T(2,:);T(7,:);T(3,:);T(8,:);T(4,:);T(5,:)>;
CBAR=CDUM*INV(TF);ADBAR=TF*AD*INV(TF);BDBAR=TF*BD;H=CBAR*BDBAR;
CLEAR SQ
SQ=<0 0 0 0 0 0 0 0;1 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0;0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0>;
SP1=<1 0 0 0 0;0 0 0 0 0;0 1 0 0 0;0 0 0 0 0;0 0 1 0 0;0 0 0 0 0;
     0 0 0 1 0;0 0 0 0 1>;
SP2=<0 0 0 0 0;1 0 0 0 0;0 0 0 0 0;0 1 0 0 0;0 0 0 0 0;0 0 1 0 0;
     0 0 0 0 0;0 0 0 0 0>;
B1ARMA=CBAR*BDBAR;
B2ARMA=CBAR*SQ*BDBAR;
GON=[B1ARMA+B2ARMA];
A1ARMA=-CBAR*SP1;
A2ARMA=-CBAR*SP2;
GODEN=[EYE(5)+A1ARMA+A2ARMA];
```

```
GO=INV(GODEN)*GON;
CHECK=-CDUM*INV(A)*B;
CLEAR CDUM SDUM
```

F.6 RLS Adaptive System Analysis Macros

The ARMA plant representations were analyzed in the discrete time domain with the fixed gain PI control law and the sampling time equal to 40 Hz with the MATRIX_x macros of the form for the ARMA implementation. This part of the appendix develops the macros used in the adaptive control law analysis using the recursive least squares algorithm.

```
//DISCRETE ANALYSIS WITH TSAMP=.025 AND ARMA MODEL REPRESENTATION
EXEC('M20');//ROUTINE TO BUILD THE GAIN MATRICES
EXEC('M21');//BUILDS THE GAIN MATRICES IN SYSTEM BUILD
EXEC('ARMA')
CLEAR B2ARMA A1ARMA A2ARMA CHECK CO TT R SD AD BD CD DD V1 VI TF T
CLEAR RN CBAR ADBAR BDBAR H SQ SP1 SP2 GON GODEN
TITLE='CRCA SIMULATION 5 INPUTS AND 5 OUTPUTS - PORTER METHOD';
DISPLAY(TITLE)
NO=0;YES=1;
//LGT SETS THE DURATION OF THE SIMULATION
INQUIRE LGT 'WHAT IS THE SIMULATION DURATION: '
LGT2=(LGT-3)/.025;
LGT1=(LGT-.5)/.025;//SETS THE RAMPING FOR SURFACE LIMITS
T1=<0:.025:LGT>';
//SUBROUTINE TO BUILD COMMAND FILES
//STEP COMMANDS
    R2A=2*.069813*<0:.025:.5>';
    R2B=<R1A;.069813*ONES(LGT1,1)>;
R2 =<0*ONES(T1),0*ONES(T1),R2B,0*ONES(T1),0*ONES(T1)>;
    R8A =2*.7854*<0:.025:.5>';
    R8A1=2*.02542*<0:.025:.5>';
    R8B =<R8A;.7854*ONES(LGT1,1)>;
    R8B1=<R8A1;.02542*ONES(LGT1,1)>;
R8 =<0*ONES(T1),0*ONES(T1),0*ONES(T1),R8B,R8B1>;
    R10A=2*.0349*<0:.025:.5>';
    R10B=<R10A;.0349*ONES(LGT1,1)>;
R10=<0*ONES(T1),R10B,0*ONES(T1),0*ONES(T1),0*ONES(T1)>;
//
CLEAR R2A R2B R8A R8A1 R8B R8B1 R10A R10B
//
G1=1;//FIXED AS A FUNCTION OF SAMPLING TIME
INQUIRE SIG1 'WHAT IS YOUR SIGMA-1 VALUE: '
INQUIRE SIG2 'WHAT IS YOUR SIGMA-2 VALUE: '
INQUIRE SIG3 'WHAT IS YOUR SIGMA-3 VALUE: '
INQUIRE SIG4 'WHAT IS YOUR SIGMA-4 VALUE: '
INQUIRE SIG5 'WHAT IS YOUR SIGMA-5 VALUE: '
INQUIRE KQ 'WHAT IS YOUR VALUE FOR KQ(M31): '
INQUIRE KP 'WHAT IS YOUR VALUE FOR KP(M42): '
INQUIRE GAM1 'WHAT IS YOUR GAMMA-1 VALUE: '
INQUIRE GAM2 'WHAT IS YOUR GAMMA-2 VALUE: '
INQUIRE GAM3 'WHAT IS YOUR GAMMA-3 VALUE: '
INQUIRE GAM4 'WHAT IS YOUR GAMMA-4 VALUE: '
```

```
INQUIRE GAM5 'WHAT IS YOUR GAMMA-5 VALUE: '
M(3,1)=KQ;M(4,2)=KP;
G=G1*EYE(5);
EXEC('M9');//COMPUTATIONAL TIME DELAY
EXEC('M12');//ACTUATOR BUILD
EXEC('M8') ;//PI CONTROLLER
EXEC('M7') ;//SYSTEM MATRIX XDOT-1 OUT
EXEC('M6') ;//PLANT WITH ACTUATORS SUPER BLOCK
EXEC('M1') ;//RLS SUPERBLOCK
EXEC('M11');//MULTI SUPERBLOCK WITH ACTUATORS
EXEC('M19');//PRE FILTERS BUILD
EXEC('M16');//MULTI SUPERBLOCK WITH PREFILTERS
BUILD,ANALYZE,MULTI1
EXEC('M18');//MENUING SYSTEM FOR INPUTS
```

Recursive Least Squares System Block

```
//M1.DAT
//THIS BLOCK WILL BUILD THE ADAPTIVE CONTROL ALGORITHM
BUILD
PLOT OFF
EDIT
ADAPT
.025
0.0
DEF,1,DYN,8,    NAME,DEL U,                 OUT,5,ENTRY,.025
DEF,2,DYN,8,    NAME,DEL Y,                 OUT,5,ENTRY,.025
DEF,3,USER,     NAME,RLS,INPUTS,31,OUT,150,STATES,0,ENTRY,1,N,1,1,1,1
CONNECT
INPUT,21,1,1,1,2,2,3,3,4,4,5,5,0,0
INPUT,21,2,6,1,7,2,8,3,9,4,10,5,0,0
INPUT,21,3,1,1,2,2,3,3,4,4,5,5,6,11,7,12,8,13,9,14,10,15,0,0
INPUT,21,3,11,21,12,22,13,23,14,24,15,25,16,26,17,27,18,28,19,29,0,0
INPUT,21,3,20,30,21,31,0,0
                  INTERNAL,1,3,1,6,2,7,3,8,4,9,5,10,0,0
                  INTERNAL,2,3,1,16,2,17,3,18,4,19,5,20,0,0
OUT,150,3,Y
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Aircraft Model With Actuators

```
//M6.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A SYSTEM MATRIX SUPER BLOCK WITH ACTUATORS
//MAKE SURE THE S MATRIX IS DEFINED
//ADDS PLANT MATRIX
PLANT1
O
DEF,2, SUPER,   NAME,PLANT,         INPUTS,6, OUT,12,ENTRY
DEF,1, SUPER,   NAME,ACT,           INPUTS,5, OUT,10,ENTRY
CONNECT
INPUT,6,1,1,1,2,2,3,3,4,4,5,5,0,0,  INTERNAL,1,2,2,1,4,2,6,3,8,4,10,5,0,0
INPUT,6,2,N,6,6,0,0

OUTPUT,22,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
OUTPUT,22,1,1,13,2,14,3,15,4,16,5,17,6,18,7,19,8,20,9,21,10,22,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Aircraft Model Without Actuators

```
//M7.DAT
BUILD
PLOT OFF
EDIT
PLANT
O
DEF,1, SUPER,              NAME,BMAT, INPUTS,6, OUT,8, ENTRY
DEF,2, ALGEBRAIC, SUM,     NAME,SUM1, INPUTS,2, OUT,8, ENTRY,2,1,1
DEF,3, DYN,NTH,   NAME,INT,           OUT,8, ENTRY,1,1,1,1,1,1,1,1,Y
DEF,4, SUPER,              NAME,CMAT, INPUTS,9, OUT,10,ENTRY
DEF,5, SUPER,              NAME,AMAT, INPUTS,9, OUT,8,ENTRY
CONNECT
INPUT,6,1,N,1,2,2,3,3,4,4,5,5,6,0,0
INPUT,6,1,N,6,1,0,0
                INTERNAL,1,2,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,0,0
                INTERNAL,2,3,Y
                INTERNAL,3,4,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,0,0
                INTERNAL,3,5,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,0,0
```

```
        INTERNAL,5,2,1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,0,0
OUTPUT,12,4,1,1,2,2,3,3,4,4,5,5,6,8,7,9,8,10,9,11,10,12,0,0
OUTPUT,12,2,1,6,2,7,0,0
//THE CONTROL INPUT FOR THE GAIN SCHEDULER IS INPUT 6
INPUT,6,4,6,1,0,0
INPUT,6,5,6,1,0,0
//OUTPUT FROM BLOCK 2 - 6 AND 7 ARE XDOT(A11 AND A12)
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

PI Controller

```
//M8.DAT
K1GAIN='Y1=U6 *U1+U7 *U2+U8 *U3+U9 *U4+U10*U5;...
        Y2=U11*U1+U12*U2+U13*U3+U14*U4+U15*U5;...
        Y3=U16*U1+U17*U2+U18*U3+U19*U4+U20*U5;...
        Y4=U21*U1+U22*U2+U23*U3+U24*U4+U25*U5;...
        Y5=U26*U1+U27*U2+U28*U3+U29*U4+U30*U5;';
K2GAIN=K1GAIN;
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A PI CONTROLLER
//THE CURRENT INPUTS ARE 5 WITH K1 AND K2
//DIMENSIONED 5 X 5
//TO CHANGE THEM FOR A 3 INPUT SYSTEM JUST
//CHANGE OUTPUTS AND INPUTS FROM 5 TO WHATEVER YOU WANT
PI
.025
.025
DEF,1, DYN,Nth,NAME,INT,OUT,5,ENTRY,1,.025,.025,.025,.025,.025,YES
DEF,2, ALGEB,     GEN,  NAME,K2,   INPUTS,30,OUT,5,ENTRY,K2GAIN
DEF,5, ALGEB,     GEN,  NAME,K1,   INPUTS,30,OUT,5,ENTRY,K1GAIN
DEF,3, ALGEBRAIC, SUM,  NAME,SUM1, INPUTS,2,OUT,5,ENTRY,2,1,1
CONNECT
INPUT,55,1,1,1,2,2,3,3,4,4,5,5,0,0
INPUT,55,5,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,0,0
INPUT,55,5,13,13,14,14,15,15,16,16,17,17,18,18,19,19,20,20,21,21,0,0
INPUT,55,5,22,22,23,23,24,24,25,25,26,26,27,27,28,28,29,29,30,30,0,0
INPUT,55,2,31,6,32,7,33,8,34,9,35,10,36,11,37,12,38,13,39,14,40,15,0,0
INPUT,55,2,41,16,42,17,43,18,44,19,45,20,46,21,47,22,48,23,49,24,0,0
```

```
INPUT,55,2,50,25,51,26,52,27,53,28,54,29,55,30,0,0
OUTPUT,5,3,Y
                         INTERNAL,1,2,1,1,2,2,3,3,4,4,5,5,0,0
                         INTERNAL,2,3,1,1,2,2,3,3,4,4,5,5,0,0
                         INTERNAL,5,3,1,6,2,7,3,8,4,9,5,10,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Plant and PI Controller

```
//M11.DAT
BUILD
PLOT OFF
EDIT
//THE FOLLOWING CREATES A MULT  SUPERBLOCK FOR SIMULATION - 5 INPUTS
//INSURE YOU HAVE CREATED THE SUPERBLOCKS PI AND APPROPRIATE PLANT
//THIS IS A SIMULATION FOR AN IRREGULAR PLANT WITH ACTUATORS
//THIS SUPERBLOCK WILL HAVE 10 OUTPUTS- OUTPUT 1-5 WILL BE COMMAND INPUTS
//OUTPUTS 5-14 WILL BE STATE VARIABLES
MULTI
0
DEF,2, SUPER,           NAME,PI,     INPUTS,55,    OUT,5,          ENTRY
DEF,4, SUPER,           NAME,PLANT1,INPUTS,6,     OUT,22,         ENTRY
DEF,3, DYN,      STATE,NAME,G,      INPUTS,5,     OUT,5,STATES,0,ENTRY,G
DEF,5, DYN,      STATE,NAME,M,      INPUTS,2,     OUT,5,STATES,0,ENTRY,M
DEF,1, SUPER,           NAME,ADAPT, INPUTS,21,    OUT,150,        ENTRY
DEF,6, ALGEBRAIC,SUM,  NAME,W,      INPUTS,2,     OUT,5,ENTRY,2,1,1
CONNECT
INPUT,17,2,1,1,2,2,3,3,4,4,5,5,0,0
INPUT,17,1,7,11,8,12,9,13,10,14,11,15,12,16,13,17,14,18,0,0
INPUT,17,1,15,19,16,20,17,21,0,0
INPUT,17,4,6,6,0,0
INTERNAL,1,2,1,6,2,7,3,8,4,9,5,10,6,11,7,12,0,0
INTERNAL,1,2,8,13,9,14,10,15,11,16,12,17,13,18,14,19,15,20,16,21,0,0
INTERNAL,1,2,17,22,18,23,19,24,20,25,21,26,22,27,23,28,24,29,25,30,0,0
INTERNAL,1,2,26,31,27,32,28,33,29,34,30,35,31,36,32,37,33,38,34,39,0,0
INTERNAL,1,2,35,40,36,41,37,42,38,43,39,44,40,45,41,46,42,47,43,48,0,0
INTERNAL,1,2,44,49,45,50,46,51,47,52,48,53,49,54,50,55,0,0
                             INTERNAL,2,3,Y
                             INTERNAL,3,4,1,1,2,2,3,3,4,4,5,5,0,0
                             INTERNAL,4,5,6,1,7,2,0,0
```

```
                           INTERNAL,4,6,1,1,2,2,3,3,4,4,5,5,0,0
                           INTERNAL,5,6,1,6,2,7,3,8,4,9,5,10,0,0
INTERNAL,4,1,14,1,16,2,18,3,20,4,22,5,0,0
INTERNAL,4,1,1,6,2,7,3,8,4,9,5,10,0,0

OUTPUT,125,4,14,1,16,2,18,3,20,4,22,5,0,0
OUTPUT,125,4,1,6,2,7,3,8,4,9,5,10,8,11,9,12,10,13,11,14,12,15,0,0
OUTPUT,125,6,1,16,2,17,3,18,4,19,5,20,0,0
OUTPUT,125,1,51,21,52,22,53,23,54,24,55,25,56,26,57,27,58,28,0,0
OUTPUT,125,1,59,29,60,30,61,31,62,32,63,33,64,34,65,35,66,36,0,0
OUTPUT,125,1,67,37,68,38,69,39,70,40,71,41,72,42,73,43,74,44,0,0
OUTPUT,125,1,75,45,76,46,77,47,78,48,79,49,80,50,81,51,82,52,0,0
OUTPUT,125,1,83,53,84,54,85,55,86,56,87,57,88,58,89,59,90,60,0,0
OUTPUT,125,1,91,61,92,62,93,63,94,64,95,65,96,66,97,67,98,68,0,0
OUTPUT,125,1,99,69,100,70,101,71,102,72,103,73,104,74,105,75,106,76,0,0
OUTPUT,125,1,107,77,108,78,109,79,110,80,111,81,112,82,113,83,114,84,0,0
OUTPUT,125,1,115,85,116,86,117,87,118,88,119,89,120,90,121,91,122,92,0,0
OUTPUT,125,1,123,93,124,94,125,95,126,96,127,97,128,98,129,99,130,100,0,0
OUTPUT,125,1,131,101,132,102,133,103,134,104,135,105,136,106,137,107,0,0
OUTPUT,125,1,138,108,139,109,140,110,141,111,142,112,143,113,144,114,0,0
OUTPUT,125,1,145,115,146,116,147,117,148,118,149,119,150,120,0,0
OUTPUT,125,4,13,121,15,122,17,123,19,124,21,125,0,0

//OUTPUTS 1-5 ARE CONTROL OUTPUTS:CL,CR,TEL,TER,RUDDER
//OUTPUTS 6-15 ARE STATE OUTPUTS:VEL,THETA,BETA,PHI,R,W,Q,P,NZ,NZPILOT
//OUTPUTS 15-20 ARE THE OUTPUTS FROM THE W SUMMER
//OUTPUTS 21-120 ARE THE PARAMETER VECTOR OUTPUT
//OUTPUTS 120-125 ARE THE SURFACE DEFLECTION RATES
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

Actuators

```
//M12.DAT
//THIS ROUTINE BUILDS THE ACTUATORS FOR THE CRCA
//FIRST ORDER MODEL OF XD=-20X+20U
BACT=20;//B MATRIX FOR THE ACTUATOR
AACT=-20;//A MATRIX FOR THE ACTUATOR
BUILD
PLOT OFF
EDIT
```

```
ACL
0
DEF,1, DYN,        STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,    NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,      3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,    NAME,LMTINT,INPUTS,1,  OUT,1,ENTRY,30,-60,1,Y
DEF,5, DYN,        STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                 INTERNAL,1,2,1,1,0,0
                           INTERNAL,2,3
                           INTERNAL,3,4
                           INTERNAL,4,5
                           INTERNAL,5,2,1,2,0,0
OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING THE RIGHT CANARD ACTUATOR
COPY,        ACL,        ACR

//BUILDING TRAILING EDGE LEFT ACTUATOR
EDIT
ATL1
0
DEF,1, DYN,        STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,    NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,      3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,    NAME,LMTINT,INPUTS,1,  OUT,1,ENTRY,30,-30,1,Y
DEF,5, DYN,        STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,                 INTERNAL,1,2,1,1,0,0
                           INTERNAL,2,3
                           INTERNAL,3,4
                           INTERNAL,4,5
                           INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILDING TRAILING EDGE RIGHT ACTUATOR
COPY,         ATL1,       ATR1
```

```
//BUILDING RUDDER ACTUATOR
EDIT
ATRUD
0
DEF,1, DYN,       STATE, NAME,B,      INPUTS,1,  OUT,1,STATES,0,ENTRY,BACT
DEF,2, ALGEBRAIC, SUM,   NAME,SUM1,   INPUTS,2,  OUT,1,ENTRY,2,1,1
DEF,3, PIECE,     3  ,   NAME,RLIM,   INPUTS,1,  OUT,1,ENTRY,100
DEF,4, DYN,NEXT,  LIM,   NAME,LMTINT,INPUTS,1,   OUT,1,ENTRY,20,-20,1,Y
DEF,5, DYN,       STATE, NAME,A,      INPUTS,1,  OUT,1,STATES,0,ENTRY,AACT
CONNECT
INPUT,1,1,               INTERNAL,1,2,1,1,0,0
                         INTERNAL,2,3
                         INTERNAL,3,4
                         INTERNAL,4,5
                         INTERNAL,5,2,1,2,0,0 OUTPUT,2,3,1,1,0,0
OUTPUT,2,4,1,2,0,0
//OUTPUT FROM BLOCK 3 - RATE LIMITER
//OUTPUT FROM BLOCK 4 - POSITION LIMITER
TOP

//BUILD SUPER BLOCK OF ACTUATORS
EDIT
ACT
0
DEF, 1, SUPER,            NAME,ACL,   INPUTS,1, OUT,2,ENTRY
DEF, 2, SUPER,            NAME,ACR,   INPUTS,1, OUT,2,ENTRY
DEF, 3, SUPER,            NAME,ATL1,  INPUTS,1, OUT,2,ENTRY
DEF, 4, SUPER,            NAME,ATR1,  INPUTS,1, OUT,2,ENTRY
DEF, 5, SUPER,            NAME,ATRUD, INPUTS,1, OUT,2,ENTRY
CONNECT
INPUT,5,1,1,1,0,0,                    OUTPUT,10,1,1,1,2,2,0,0
INPUT,5,2,2,1,0,0,                    OUTPUT,10,2,1,3,2,4,0,0
INPUT,5,3,3,1,0,0,                    OUTPUT,10,3,1,5,2,6,0,0
INPUT,5,4,4,1,0,0,                    OUTPUT,10,4,1,7,2,8,0,0
INPUT,5,5,5,1,0,0,                    OUTPUT,10,5,1,9,2,10,0,0

TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR BACT AACT
```

<u>Aircraft Model With PI Controller and Pre-Filter</u>

```
//M16.DAT
//THIS EXECUTABLE FILE BUILDS A MULTI SUPERBLOCK WITH PRE FILTER
//ALSO BUILDS NORMAL ACCELERATION AT PILOT STATION OUTPUT
BUILD
PLOT OFF
EDIT
MULTI1
0
DEF,3,  SUPER,      NAME,MULTI,         INPUTS,17,  OUT,125,  ENTRY
DEF,2,  ALGEBRAIC,  SUM,NAME,ER,        INPUTS,2,   OUT,5,ENTRY,2,1,-1
DEF,1,  SUPER,      NAME,PREF,          INPUTS,5,   OUT,5,    ENTRY
DEF,5,  SUPER,      NAME,NZMAT,         INPUTS,5,   OUT,1,ENTRY
DEF,4,  ALGEBRAIC,  SUM,NAME,NZ,        INPUTS,2,   OUT,1,ENTRY,2,1,1
CONNECT
INPUT,17,1,1,1,2,2,3,3,4,4,5,5,0,0
INPUT,17,3,N,6,6,7,7,8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,0,0
INPUT,17,3,N,16,16,17,17,0,0
INPUT,17,5,6,1,0,0
                    INTERNAL,1,2,1,1,2,2,3,3,4,4,5,5,0,0
                    INTERNAL,2,3,1,1,2,2,3,3,4,4,5,5,0,0
                    INTERNAL,3,2,16,6,17,7,18,8,19,9,20,10,0,0
                    INTERNAL,5,4,1,1,0,0
                    INTERNAL,3,4,14,2,0,0
                    INTERNAL,3,5,1,2,2,3,3,4,4,5,0,0
OUT,125,3,N,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,0,0
OUT,125,3,N,12,12,13,13,14,14,0,0
OUT,125,4,1,15,0,0
OUT,125,1,1,16,2,17,3,18,4,19,5,20,0,0
OUT,125,3,N,21,21,22,22,23,23,24,24,25,25,26,26,27,27,28,28,29,29,0,0
OUT,125,3,N,30,30,31,31,32,32,33,33,34,34,35,35,36,36,37,37,38,38,0,0
OUT,125,3,N,41,41,42,42,43,43,44,44,45,45,46,46,47,47,48,48,49,49,0,0
OUT,125,3,N,51,51,52,52,53,53,54,54,55,55,56,56,57,57,58,58,59,59,0,0
OUT,125,3,N,61,61,62,62,63,63,64,64,65,65,66,66,67,67,68,68,69,69,0,0
OUT,125,3,N,71,71,72,72,73,73,74,74,75,75,76,76,77,77,78,78,79,79,0,0
OUT,125,3,N,81,81,82,82,83,83,84,84,85,85,86,86,87,87,88,88,89,89,0,0
OUT,125,3,N,91,91,92,92,93,93,94,94,95,95,96,96,97,97,98,98,99,99,0,0
OUT,125,3,N,101,101,102,102,103,103,104,104,105,105,106,106,107,107,0,0
OUT,125,3,N,109,109,110,110,111,111,112,112,113,113,114,114,115,115,0,0
OUT,125,3,N,117,117,118,118,119,119,120,120,0,0
OUT,125,3,N,121,121,122,122,123,123,124,124,125,125,0,0
OUT,125,3,N,39,39,40,40,50,50,60,60,70,70,80,80,90,90,100,100,0,0
OUT,125,3,N,108,108,116,116,0,0
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
```

```
//M18.DAT
ERASE;CLEAR Y
INQUIRE TYPE 'WHAT FLIGHT MANEUVER WOULD YOU LIKE: '
INQUIRE FLAG1 'WHAT INITIAL FAILURE OR FLIGHT MODE DO YOU WANT: '
INQUIRE FLAG2 'WHAT FAILURE OR FLIGHT MODE AT 2 SECONDS: '
INQUIRE FLAG3 'WHAT IS YOUR FORGETTING FACTOR: '
T2=<0:.025:2>';
T3=(LGT-2)/.025;
FAIL=<FLAG1*ONES(T2);FLAG2*ONES(T3,1)>;
FORG=FLAG3*ONES(T1);
CLEAR FLAG1 FLAG2 FLAG3 T2 T3
TYPE(:,6)=FAIL;//GAIN SCHEDULING FOR CONVERGENCE TEST
TYPE(:,7) =SIG1*ONES(T1);//SIGMA 1 INPUT
TYPE(:,8) =SIG2*ONES(T1);//SIGMA 2 INPUT
TYPE(:,9) =SIG3*ONES(T1);//SIGMA 3 INPUT
TYPE(:,10)=SIG4*ONES(T1);//SIGMA 4 INPUT
TYPE(:,11)=SIG5*ONES(T1);//SIGMA 5 INPUT
TYPE(:,12)=GAM1*ONES(T1);//GAMMA 1 INPUT
TYPE(:,13)=GAM2*ONES(T1);//GAMMA 2 INPUT
TYPE(:,14)=GAM3*ONES(T1);//GAMMA 3 INPUT
TYPE(:,15)=GAM4*ONES(T1);//GAMMA 4 INPUT
TYPE(:,16)=GAM5*ONES(T1);//GAMMA 5 INPUT
TYPE(:,17)=FORG;//FORGETTING FACTOR
CLEAR SIG1 SIG2 SIG3 SIG4 SIG5
CLEAR GAM1 GAM2 GAM3 GAM4 GAM5 FORG
SIM('IALG')
STIFF SYSTEM SOLVER
Y=HSIM(T1,TYPE);
PLOT(T1,Y(:,[6 16]),'XLABEL/TIME IN SECONDS/,...
YLABEL/FT-SEC/,TITLE/        VELOCITY VS VELOCITY COMMAND/,UPPER LEFT');
PLOT(T1,Y(:,[7 17])*57.3,'XLABEL/TIME IN SECONDS/,...
YLABEL/DEGREES/,TITLE/     BETA VS BETA-COMMAND/,UPPER RIGHT,KEEP');
PLOT(T1,Y(:,[8 18])*57.3,'XLABEL/TIME IN SECONDS/,...
YLABEL/DEGREES/,TITLE/     THETA VS THETA-CMD/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,[9 19])*57.3,'XLABEL/TIME IN SECONDS/,...
YLABEL/DEGREES/,TITLE/         PHI VS PHI-COMMMAND/,LOWER RIGHT,KEEP');
PLOT(T1,Y(:,[10 20])*57.3,'XLABEL/TIME IN SECONDS/,...
YLABEL/DEG-SEC/,TITLE/       YAW RATE VS YAW RATE COMMAND/,UPPER');
PLOT(T1,Y(:,5),'XLABEL/TIME IN SECONDS/,...
```

```
YLABEL/DEGREES/,TITLE/          RUDDER DEFLECTION/,LOWER,KEEP');
PLOT(T1,Y(:,1),'XLABEL/TIME IN SECONDS/,...
YLABEL/DEGREES/,TITLE/         CANARD LEFT DEFLECTION/,UPPER LEFT');
PLOT(T1,Y(:,2),'XLABEL/TIME IN SECONDS/,...
YLABEL/DEGREES/,TITLE/        CANARD RIGHT DEFLECTION/,UPPER RIGHT,KEEP');
PLOT(T1,Y(:,3),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES/,...
TITLE/        TRAILING EDGE LEFT DEFLECTION/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,4),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES/,...
TITLE/        TRAILING EDGE RIGHT DEFLECTION/,LOWER RIGHT,KEEP');
PLOT(T1,Y(:,121),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES-SEC/,...
TITLE/LEFT CANARD DEFLECTION RATE/,UPPER LEFT');
PLOT(T1,Y(:,122),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES-SEC/,...
TITLE/RIGHT CANARD DEFLECTION RATE/,UPPER RIGHT,KEEP');
PLOT(T1,Y(:,123),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES-SEC/,...
TITLE/LEFT TRAILING EDGE DEFLECTION RATE/,LOWER LEFT,KEEP');
PLOT(T1,Y(:,124),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES-SEC/,...
TITLE/RIGHT TRAILING EDGE DEFLECTION RATE/,LOWER RIGHT,KEEP');
PLOT(T1,Y(:,125),'XLABEL/TIME IN SECONDS/ YLABEL/DEGREES-SEC/,...
TITLE/RUDDER DEFLECTION RATE/');
RETURN
```

Low-Pass Filter

```
//M19.DAT
//THE FOLLOWING EXECUTABLE ROUTINE BUILDS A LOW PASS PREFILTER
PRF=<-10,1;10,0>;
BUILD
PLOT OFF
EDIT
PREF
O
DEF,1, DYN,  STATE,NAME,  PRE1,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,2, DYN,  STATE,NAME,  PRE2,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,3, DYN,  STATE,NAME,  PRE3,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,4, DYN,  STATE,NAME,  PRE4,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
DEF,5, DYN,  STATE,NAME,  PRE5,   INPUTS,1,  OUT,1, STATES,1,ENTRY,PRF,Y
CONNECT
INPUT,5,1,1,1,0,0,                           OUT,5,1,1,1,0,0
INPUT,5,2,2,1,0,0,                           OUT,5,2,1,2,0,0
INPUT,5,3,3,1,0,0,                           OUT,5,3,1,3,0,0
INPUT,5,4,4,1,0,0,                           OUT,5,4,1,4,0,0
INPUT,5,5,5,1,0,0,                           OUT,5,5,1,5,0,0
```

```
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR PRF
```

Gain Scheduling

```
//M20.DAT
//THIS FILE BUILDS THE GAIN MATRICES USED IN THE MATRIXX SIMULATION
//THE PROGRAM WILL CALL EACH OF THE MATRICES AND BUILD THE ONE GAIN
//MATRIX A,B,C
FLAG=0;FLAG1=0;//INITIALIZATION OF VARIABLES

EXEC('ACMENTRY');//BASIC ACMENTRY MATRICES
AMAT1=A;
BMAT1=B;
CMAT1=C;
NZMAT1=NZM;
//******************************************
EXEC('ACM25CL');//25 PERCENT LOE LEFT CANARD
AMAT2=A;
BMAT2=B;
CMAT2=C;
NZMAT2=NZM;
//*******************************************
EXEC('ACM50CL');//50 PERCENT LOE LEFT CANARD
AMAT3=A;
BMAT3=B;
CMAT3=C;
NZMAT3=NZM;
//*******************************************
EXEC('ACM30TL');//30 PERCENT LOE LEFT TRAILING SURFACE
AMAT4=A;
BMAT4=B;
CMAT4=C;
NZMAT4=NZM;
//*******************************************
EXEC('ACMEXIT');//ACMEXIT
AMAT5=A;
BMAT5=B;
CMAT5=C;
NZMAT5=NZM;
```

```
//*******************************************
EXEC('TFTA');//TFTA BASIC FLIGHT CONDITION
AMAT6=A;
BMAT6=B;
CMAT6=C;
NZMAT6=NZM;
//*******************************************
EXEC('TFTA25CL');//TFTA 25 PERCENT CANARD LEFT LOE
AMAT7=A;
BMAT7=B;
CMAT7=C;
NZMAT7=NZM;
//*******************************************
EXEC('TFTA50CL');//TFTA 50 PERCENT CANARD LEFT LOE
AMAT8=A;
BMAT8=B;
CMAT8=C;
NZMAT8=NZM;
//*******************************************
EXEC('TFTA30TL');//TFTA 30 PERCENT TRAILING EDGE LEFT LOE
AMAT9=A;
BMAT9=B;
CMAT9=C;
NZMAT9=NZM;
//*******************************************
AMAT=<AMAT1,AMAT2,AMAT3,AMAT4,AMAT5,AMAT6,AMAT7,AMAT8,AMAT9>;
BMAT=<BMAT1,BMAT2,BMAT3,BMAT4,BMAT5,BMAT6,BMAT7,BMAT8,BMAT9>;
CMAT=<CMAT1,CMAT2,CMAT3,CMAT4,CMAT5,CMAT6,CMAT7,CMAT8,CMAT9>;
NZMAT=<NZMAT1,NZMAT2,NZMAT3,NZMAT4,NZMAT5,NZMAT6,NZMAT7,NZMAT8,NZMAT9>;
CLEAR AMAT1 AMAT2 AMAT3 AMAT4 AMAT5 AMAT6 AMAT7 AMAT8 AMAT9
CLEAR BMAT1 BMAT2 BMAT3 BMAT4 BMAT5 BMAT6 BMAT7 BMAT8 BMAT9
CLEAR CMAT1 CMAT2 CMAT3 CMAT4 CMAT5 CMAT6 CMAT7 CMAT8 CMAT9
CLEAR NZMAT1 NZMAT2 NZMAT3 NZMAT4 NZMAT5 NZMAT6 NZMAT7 NZMAT8 NZMAT9
//*******************************************
//EXECUTING THE FOLLOWING COMMAND FILE WILL SET THE INITIAL
//MATRICES FOR THE GAIN CONTROL LAW FOR THE FIXED GAIN CASES
TITLE='CHOOSE THE BASIC FLIGHT CONDITION FOR WHICH GAINS ARE CALCULATED';
DISPLAY(TITLE);CLEAR TITLE
TITLE='ACMENTRY=1        ACMEXIT=2         TFTA=3';
DISPLAY(TITLE);CLEAR TITLE
INQUIRE FLAG 'ENTER FLIGHT CONDITION: '
IF FLAG=1, EXEC('ACMENTRY')
IF FLAG=2, EXEC('ACMEXIT')
IF FLAG=3, EXEC('TFTA')
TITLE ='ACMENTRY=1';
```

```
TITLE1='ACM25CL =2';
TITLE2='ACM50CL =3';
TITLE3='ACM30TL =4';
TITLE4='ACMEXIT =5';
TITLE5='TFTA    =6';
TITLE6='TFTA25CL=7';
TITLE7='TFTA50CL=8';
TITLE8='TFTA30TL=9';
DISPLAY(TITLE)
DISPLAY(TITLE1)
DISPLAY(TITLE2)
DISPLAY(TITLE3)
DISPLAY(TITLE4)
DISPLAY(TITLE5)
DISPLAY(TITLE6)
DISPLAY(TITLE7)
DISPLAY(TITLE8)
CLEAR TITLE TITLE1 TITLE2 TITLE3 TITLE4 TITLE5 TITLE6 TITLE7 TITLE8
//*****************************************************************
```

Gain Scheduling Matrix Superblocks

```
//M21.DAT
//THIS COMMAND FILE BUILDS THE SUPERBLOCK GAIN MATRICES AMAT,BMAT,CMAT
P=<.5,1.5,2.5,3.5,4.5,6.5,7.5,8.5,9.5>;
BUILD,PLOT OFF
EDIT
AMAT
O
DEF,2,NEXT, LOGIC,6, NAME,AGAIN,INPUTS,9,OUT,8,ENTRY,P,AMAT,.01
CONNECT
INPUT,9,2,Y,                    OUT,8,2,Y
TOP
//*************************************************************
EDIT
BMAT
O
DEF,2,NEXT, LOGIC,6, NAME,BGAINS,INPUTS,6,OUT,8,ENTRY,P,BMAT,.01
CONNECT
INPUT,6,2,Y,                 OUT,8,2,Y
TOP
```

```
//**************************************************************
EDIT
CMAT
O
DEF,2,NEXT, LOGIC,6, NAME,CGAINS,INPUTS,9,OUT,10,ENTRY,P,CMAT,.01
CONNECT
INPUT,9,2,Y,              OUT,10,2,Y
TOP
//**************************************************************
EDIT
NZMAT
O
DEF,2,NEXT, LOGIC,6, NAME,NZGAIN,INPUTS,5,OUT,1,ENTRY,P,NZMAT,.01
CONNECT
INPUT,5,2,Y,              OUT,1,2
TOP
//THE FOLLOWING STATEMENT RETURNS YOU TO MATRIXX
MAT
CLEAR P
```

ARMA Calculation Macro

```
//THE FOLLOWING PROGRAM CALCULATES THE ARMA MODEL FOR THE PLANTS
LONG E
CLEAR CO TT R SD
CDUM=<C1 C2>;
SDUM=<A,B;CDUM,0*EYE(5)>;
SD=DISC(SDUM,8,.025);
[AD BD CD DD]=SPLIT(SD,8);
CO=<CDUM*INV(AD)>;FOR I=2:8,CO=<CO;CDUM*INV(AD)**I>;END;
V1=0;CR=0;TT=<CO(1,:)>;R=TT;FOR I=2:16,R=<TT;CO(I,:)>;...
RN=RANK(R);IF RN=I-CR THEN TT=R;ELSEIF RN<I-CR THEN R=TT;...
VI=<V1;I>;V1=VI;CR=CR+1;END,END,T=R;CO;
TF=<T(1,:);T(6,:);T(2,:);T(7,:);T(3,:);T(8,:);T(4,:);T(5,:)>;
CBAR=CDUM*INV(TF);ADBAR=TF*AD*INV(TF);BDBAR=TF*BD;H=CBAR*BDBAR,;
CLEAR SQ
SQ=<0 0 0 0 0 0 0 0;1 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0;0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0>;
SP1=<1 0 0 0 0;0 0 0 0 0;0 1 0 0 0;0 0 0 0 0;0 0 1 0 0;0 0 0 0 0;
     0 0 0 1 0;0 0 0 0 1>;
SP2=<0 0 0 0 0;1 0 0 0 0;0 0 0 0 0;0 1 0 0 0;0 0 0 0 0;0 0 1 0 0;
     0 0 0 0 0;0 0 0 0 0>;
```

```
B1ARMA=CBAR*BDBAR;
B2ARMA=CBAR*SQ*BDBAR;
GON=[B1ARMA+B2ARMA];
A1ARMA=-CBAR*SP1;
A2ARMA=-CBAR*SP2;
GODEN=[EYE(5)+A1ARMA+A2ARMA];
GO=INV(GODEN)*GON
CHECK=-CDUM*INV(A)*B
CLEAR CDUM SDUM
```

Recursive Least Squares Implementation

```
C----------------------------------------------------------------
C|                                                              |
C| THIS PROGRAM PROVIDES A RECURSIVE LEAST SQUARES ESTIMATION   |
C| OF THE PARAMETERS FOR THE CRCA 5 X 5 A AND B COEFFICIENTS.   |
C|                LAST REVISION 5 AUGUST 88                     |
C----------------------------------------------------------------
      SUBROUTINE UPDUSR (INFO,NUMBER,T,U,NU,X,XDOT,NX,Y,NY,
     +                   RP,IP)
      DOUBLE PRECISION T,U(1),X(1),Y(1),XDOT(1),RP(1)
      INTEGER          MAXNUM, INFO(4), IP(1)
      CHARACTER*2 CNUM
      DATA             MAXNUM / 1 /
C
      IF (NUMBER.GT.MAXNUM) THEN
          INFO(1)=-2
          WRITE(CNUM,111) NUMBER
 111      FORMAT(I2)
          CALL MATWR(' ')
          CALL MATWR
     +       ('SIM_ERROR: NOT ABLE TO UPDATE USER FUNCTION'//CNUM)
          RETURN
      ENDIF
      CALL USR01 (INFO,T,U,NU,X,XDOT,NX,Y,NY,RP,IP)
      RETURN
      END
C-----------------------------------------------------------------
C|  START OF USER SUBROUTINE FOR THE RECURSIVE LEAST SQUARES     |
C|    ESTIMATION AND GAIN MATRIX CALCULATIONS FOR THE ARMA       |
C|                    MODEL REPRESENTATION                       |
C-----------------------------------------------------------------
      SUBROUTINE USR01(INFO,T,U,NU,X,XDOT,NX,Y,NY,RP,IP)
```

```
      DOUBLE PRECISION T,U(1),X(1),XDOT(1),Y(1),RP(1),
     +                 XNPLUS(100,5),GAMD1(5,5),XNPLUST(5,100),
     +                 XNPLUSPN(5,100),PN(100,100),GAMMA(5,5),
     +                 FORGET(5,5),PNXNPLUS(100,5),
     +                 GON(5,5),HINV(5,5),A1ARMA(5,5),
     +                 A2ARMA(5,5),B1ARMA(5,5),B2ARMA(5,5),
     +                 K1(5,5),K2(5,5),PND2(100,100),
     +                 THED2(5),THED4(100),
     +                 THETA(100),EYE5(5,5),GAMMAI(5,5),THED3(5),
     +                 SIGMA(5),GO(5,5),GODEN(5,5),PIT(5),
     +                 GODENI(5,5),PNPLUS(100,100),THETA1(100),
     +                 PXPLGAM(100,5),WKAREA(40),GOI(5,5)
      INTEGER          IP(1), INFO(4),N,IA,IDGT,IER,I,J,K,L
      LOGICAL          INIT,STATE,OUTPUT

      INIT  = INFO(2).NE.0
      STATE = INFO(3).NE.0
      OUTPUT= INFO(4).NE.0
      IF (STATE .OR. (.NOT. OUTPUT)) THEN
      GOTO 999
      ENDIF
      IDGT=10     !PARAMETERS FOR IMSL LIBRARY
      N=5         !PARAMETERS FOR IMSL LIBRARY
      IA=5        !PARAMETERS FOR IMSL LIBRARY
C------------------------------------------------------------
C|   CHECK FOR INITIALIZATION OF THE THETA(0) AND P(0)      |
C|-----------------------------------------------------------
      TSTART=.050                                !3 PERIODS
      IF (TSTART.GT.T) THEN
         OPEN(UNIT=102,FILE='THETANOM.RLS',STATUS='OLD')!THETA
         DO I=1,100
            READ(102,*)THETA(I)
         END DO
         CLOSE(102)
         OPEN(UNIT=103,FILE='PONOM.RLS',STATUS='OLD')   !PN
         DO I=1,100
            DO J=1,100
               READ(103,*)PN(I,J)
            END DO
         END DO
         CLOSE(103)
      GOTO 600 !INSURES FULLY POPULATED XNPLUS
      ENDIF
C---------------------------------------------------------
C|         CALCULATION OF NEW PARAMETER VECTOR           |
```

```
C------------------------------------------------------------
      CALL MMULT1 (5,100,1,XNPLUST,THETA,THED2)
      DO I=1,5
         THED3(I)   =  U(I+10)-THED2(I)
      END DO
      CALL MMULT1 (100,5,1,PXPLGAM,THED3,THED4)
      DO J=1,100
        THETA1(J) = THETA(J) + THED4(J)
      END DO
C-------------------------------------------------------------
C| UPDATE PARAMETER VECTOR FOR NEXT ITERATION                 |
C-------------------------------------------------------------
      DO I=1,100
         THETA(I)=THETA1(I)
      END DO
C--------------------------------------------------------------
C|     THE RECURSIVE UPDATE EQUATION STARTS HERE.             |
C--------------------------------------------------------------
C--------------------------------------------------------------
C| UPDATE XNPLUS WITH THE OUTPUT FOR NEXT ITERATION           |
C--------------------------------------------------------------
600   K=1                            !COUNTER FOR UPDATE MATRIX
      DO J=75,95,5
  DO I=1,5
            XNPLUS(J+I,K) = U(I+5)   !U(I+5)--U(T-2)
          END DO
      K=K + 1
      END DO
C---------------------------------------------------------------
      K=1                            !COUNTER FOR UPDATE MATRIX
      DO J=50,70,5
  DO I = 1,5
            XNPLUS(J+I,K)=U(I)       !U(I)--U(T-1)
          END DO
      K=K + 1
      END DO
C-------------------------------------------------------------
      K=1
      DO J=25,45,5
         DO I=1,5
            XNPLUS(J+I,K) = -U(I+15)  !UPDATES Y(T-2)
         END DO
      K=K+1
      END DO
      K=1
```

```
C-------------------------------------------------------------
      DO J=0,20,5
         DO I=1,5
            XNPLUS(J+I,K)=  -U(I+10)  ! UPDATES Y(T-1)
         END DO
      K=K+1
      END DO
C--------------------------------------------------------------
C| CALCULATION OF P(K) * XN(K+ 1)                             |
C--------------------------------------------------------------
      CALL MMULT  (100,100,5,PN,XNPLUS,PNXNPLUS)
C-------------------------------------------------------------
C|                CALCULATE XNPLUST * PN                     |
C-------------------------------------------------------------
C--------------------------------------------------------------
C| THE FOLLOWING LOOP UPDATES THE FORGETTING FACTOR MATRIX    |
C--------------------------------------------------------------
      DO I=1,5
         DO J=1,5
            IF(I.EQ.J)THEN
              FORGET(I,J)=U(31)
            ELSE
              FORGET(I,J)=0
            END IF
         END DO
      END DO
      DO I=1,5
         DO J=1,100
            XNPLUST(I,J) = XNPLUS(J,I)
         END DO
      END DO
      CALL MMULT (5,100,100,XNPLUST,PN,XNPLUSPN)
C------------------------------------------------------------
C|                CALCULATE X(N+1)T*PN*X(N+1)               |
C------------------------------------------------------------
      CALL MMULT (5,100,5,XNPLUSPN,XNPLUS,GAMD1)
C------------------------------------------------------------
C|                   CALCULATE GAMMA(N+1)                   |
C------------------------------------------------------------
      DO I=1,5
         DO J=1,5
            GAMMA(I,J) = FORGET(I,J) + GAMD1(I,J)
         END DO
      END DO
C------------------------------------------------------------
```

```
C|     COMPUTE GAMMA(N+1) INVERSE USING IMSL ROUTINE         |
C------------------------------------------------------------
      CALL LINV2F (GAMMA,N,IA,GAMMAI,IDGT,WKAREA,IER)
      CALL MMULT  (100,5,5,PNXNPLUS,GAMMAI,PXPLGAM)
C------------------------------------------------------------
C|     CALCULATE PNPLUS, UPDATE OF COVARIANCE MATRIX         |
C------------------------------------------------------------
      CALL MMULT  (100,5,100,PXPLGAM,XNPLUSPN,PND2)
C===========================================================
      DO J=1,100
         DO K=1,100
            PNPLUS(J,K)=PN(J,K)-PND2(J,K)
         END DO
      END DO
C-----------------------------------------------------------
C| UPDATE COVARIANCE MATRIX AND PREPARE FOR NEXT ITERATION |
C|----------------------------------------------------------
      DO I=1,100
         DO J=1,100
            PN(I,J) = PNPLUS(I,J)
         END DO
      END DO
C--------------------------------------------------------------
C|                                                            |
C| THIS CONCLUDES THE BASIC CALCULATION OF THE LEAST SQUARES  |
C|  ESTIMATE OF THE PARAMETER VECTOR THETA.  NEXT THE GAIN    |
C|            MATRICES WILL BE CALCULATED                     |
C|                                                            |
C--------------------------------------------------------------
C--------------------------------------------------------------
C|                                                            |
C|      CALCULATION OF THE IDENTITY MATRIX FOR  5 X 5         |
C--------------------------------------------------------------
      DO I=1,5
         DO J=1,5
            IF(I.EQ.J)THEN
               EYE5(I,J)=1
            ELSE
               EYE5(I,J)=0
            END IF
         END DO
      END DO
C------CALCULATION OF H AND H INVERSE------------------------
      K=0
      DO I=1,5
```

```
        DO J=1,5
           A1ARMA(I,J)=THETA(K + J)
           A2ARMA(I,J)=THETA(25 +K + J)
           B1ARMA(I,J)=THETA(50+K+J)
           B2ARMA(I,J)=THETA(75+K+J)
           GON(I,J)   =B1ARMA(I,J) + B2ARMA(I,J)
           GODEN(I,J) =EYE5(I,J) + A1ARMA(I,J) + A2ARMA(I,J)
        END DO
      K=K + 5
      END DO
C==============================================================
      CALL LINV2F (B1ARMA,N,IA,HINV,IDGT,WKAREA,IER)
      CALL LINV2F (GODEN,N,IA,GODENI,IDGT,WKAREA,IER)
C--------------------------------------------------------------
C|              INITIALIZE SIGMA MATRIX                        |
C--------------------------------------------------------------
      DO I=1,5
         SIGMA(I)=U(I+20)
      END DO
C--------------------------------------------------------------
C|              INITIALIZE PI MATRIX                           |
C--------------------------------------------------------------
      DO I=1,5
        PIT(I)=U(25+I)
      END DO
C--------------------------------------------------------------
C|   CALCULATION OF G(O) AND INV(GO)                           |
C--------------------------------------------------------------
      CALL MMULT (5,5,5,GODENI,GON,GO)
      CALL LINV2F (GO,N,IA,GOI,IDGT,WKAREA,IER)
C--------------------------------------------------------------
C|   CALCULATION OF K1 AND K2 FOR GAIN                         |
C--------------------------------------------------------------
      DO I=1,5
        DO J=1,5
          K1(I,J)= HINV(I,J)*SIGMA(J)
          K2(I,J)= GOI(I,J)*PIT(J)
        END DO
      END DO
C---------------------------------------------------------------
C| THE FOLLOWING ROUTINES WILL PREPARE THE OUTPUTS OF THE USER |
C| BLOCK. Y(1) - Y(25)=K1, Y(26)-Y(50)=K2,Y(51)-Y(150)=THETA   |
C|--------------------------------------------------------------
      L=25
      K=0
```

```
      DO I=1,5
         DO J=1,5
            Y(J+K)=K1(I,J)
            Y(J+L)=K2(I,J)
         END DO
      K=K+5
      L=L+5
      END DO
      DO I=1,100
         Y(I+50)=THETA(I)
      END DO
  999 RETURN
      END
C-----------------------------------------------------------------
C|     SUBROUTINE MMULT FOR THE MATRIX MULTIPY-SIZE > 1          |
C-----------------------------------------------------------------
      SUBROUTINE MMULT (MATARW,MATACL,MATABCL,MATA,MATB,MATC)
      DOUBLE PRECISION MATC(MATARW,MATABCL),MATA(MATARW,MATACL),
     +                 MATB(MATACL,MATABCL)
      INTEGER MATARW,MATACL,MATABCL
      DO I=1,MATARW
         DO K=1,MATABCL
            SUM=0              !RESET PRODUCT SUM
            DO J=1,MATACL
               SUM=SUM + MATA(I,J)*MATB(J,K)
            END DO
         MATC(I,K)=SUM
         END DO
      END DO
      RETURN
      END
C-----------------------------------------------------------------
C|   MATRIX MULTIPLY ROUTINE FOR MATRICES WITH A COLUMN          |
C-----------------------------------------------------------------
      SUBROUTINE MMULT1 (MATARW,MATACL,MATABCL,MATA,MATB,MATC)
      DOUBLE PRECISION MATC(MATARW),MATA(MATARW,MATACL),
     +                 MATB(MATACL)
      INTEGER MATARW,MATACL,MATABCL
      DO I=1,MATARW
            SUM=0              !RESET PRODUCT SUM
            DO J=1,MATACL
               SUM = SUM + MATA(I,J)*MATB(J)
            END DO
         MATC(I) = SUM
      END DO
```

```
RETURN
END
```

Bibliography

1. Astrom, K. J., "Theory and Applications of Adaptive Control-A Survey," *Automatica*, Vol 19, No.5, 1983, pp 471-486.
2. Barfield, F.A., *Multivariable Control Laws for the AFTI/F-16.* MS Thesis, AFIT/GE/EE-83S-4. School of Engineering, Wright-Patterson AFB, OH, December 1983.
3. Barron Roger L. and P. Hess, "CRCA Linearized Equations of Motion and Closed-Loop Simulations." *Interim Technical Report*, Purchase Order 5035, F33615-84-C-3609, September 10, 1986.
4. Blakelock, J. *Automatic Control of Aircraft and Missiles.* New York: John Wiley and Sons, Inc., 1965.
5. Bokor, J. and L. Keviczky, "ARMA Canonical Forms Obtained from Constructibility Invariants." *International Journal of Control*, Vol 45, No 3, pp 861-873, 1987.
6. Bradshaw, A. and Porter, B. "Singular Perturbation Methods in the Design of Tracking-Systems Incorporating Fast-Sampling Error-Actuated Controllers," *International Journal of Systems Science*, Vol 12, pp 1207 - 1220. October 1981.
7. Courtheyn, T. L., *Multivariable Control Law Design for the X-29 Aircraft.* MS Thesis, AFIT/GE/EE-84D-21. School of Engineering, Wright-Patterson AFB, OH, December 1984.
8. D'Azzo, J. and C. Houpis. *Linear Control Systems Analysis and Design.* Chapter 20 (Third Edition). New York: McGraw-Hill, 1981.
9. Eslinger, R. A., *Multivariable Control Law Development for the AFTI F-16.* MS Thesis, AFIT/GE/EE-84S-4. School of Engineering, Wright-Patterson AFB, OH, December 1984.
10. Hofmann, L.G. "Technique for Analysis of Digital Control Systems",General Electric Company, Avionic and Electronic Systems Division. Binghampton, New York.
11. Houpis, Constantine H. and Gary Lamont.*Digital Control Systems.* McGraw-Hill, 1981.
12. Hsia, T.C., *System Identification - Least Squares Method*, D.C. Heath and Company, Lexington, Massachusetts, 1977.
13. Issermann, R., "Parameter-Adaptive Control Algorithms - A Tutorial," *Automatica*, Vol 18, No. 5, 1982, pp 513-528.
14. Ljung, L., and T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Mass. USA, 1983.

15. "MATRIX$_x$ User's Guide,"Integrated Systems Inc., Palo Alto, California, 1982

16. Pineiro, L.A., *Parameter Adaptive Model Following for In-Flight Simulation.* MS Thesis, AFIT/GE/ENG/87M. School of Engineering, Wright-Patterson AFB, OH, December 1987.

17. Porter,B. "Design of Set-Point Tracking and Disturbance Rejection Controllers for Unknown Multivariable Plants," *IEE Workshop on the Theory and Application of Self-Tuning and Adaptive Control*, Oxford England, March 1981. Report No. VSAME/DL/101/81. Salford, England: University of Salford, January 1981.

18. Porter. B. and R.N. Fripp, "Design of Fail-Safe Adaptive Digital Set-Point Tracking Controllers for Multivariable Plants," *Proceeding IFAC Conference.* 1986.

19. Porter. B., and A.H. Jones. "Design of Adaptive Digital Set-Point Tracking Controllers Incorporating Recursive Step-Response Matrix Identifiers for Multivariable Plants." *Fourth IASTED International Symposium on Modelling, Identification and Control.* Grindelwald, Switzerland. 19-22 February 1985.

20. Porter, B., A. Managanas, and T. Manganas, "Design of Digital Model-Following Flight-Mode Control Systems for High Performance Aircraft," *AIAA Guidance, Navigation, and Control* Conference, Minneapolis, Minnesota, 15-17 August 1988.

21. Porter, B., A. Managanas, and T. Manganas, "Design of Fast Non-Interacting Digital Flight-Mode Control Systems for High Performance Aircraft," *AIAA Guidance, Navigation, and Control* Conference, Snowmass, Colorado, August 1985.

22. Porter. B., and others, "Design of Adaptive Direct Digital Flight-Mode Control Systems Incorporating Recursive Step-Response Matrix Identifies for High Performance Aircraft." *Proceedings of the IEEE National Aerospace Conference.* Dayton, OH. May 1986.

23. Rosenbrok, H. H., "The Zeros of a System." *International Journal of Control*, Vol 18, No. 2, pp 297-299, 1973.

24. Sheldon, Stuart N., *Multivariable Control Law Design for a STOL F-15 Derivative in the Landing Configuration Using a PID Controller*, MS Thesis, GE/ENG/86D-24, School of Engineering, Air Force Institute of Technology (AU) Wright-Patterson AFB, Ohio, 1986.

25. Velez, Julio E., *Multivariable Control Law Design for the AFTI/16 With a Failed Control Surface Using a Parameter-Adaptive Controller*, MS Thesis, GE/ENG/87D-69, School of Engineering, Air Force Institute of Technology (AU) Wright-Patterson AFB, Ohio, 1987.

26. Weinstein, Warren, et. al., "*Control Reconfigurable Combat Aircraft Development Phase 1 - R and D Design Evaluation,*" AFWAL-TR-87-3011, Wright Patterson AFB, OH, May 1987.

27. Wittenmark, Björn, and Karl J. Astrom, "Practical Issues in the Implementation of Self-Tuning Control," *Automatica*, Vol 20, No.5, 1984, pp 595-605.

Vita

Captain Daryl Hammond attended California State University, Sacramento, and received a Bachelor of Science degree in Electrical and Electronics Engineering. He is a registered Professional Electrical Engineer in the State of California and holds an electrical contracting license from that state. His previous assignment was at the Satellite Control Facility at Onizuka Air Force Station where he was an Interface Control Engineer for the Data Systems Modernization Program Office. In May 1987 he was assigned to the School of Engineering, Air Force Institute of Technology.

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A202 697

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/88D-14	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Flight Dynamics Lab 8b. OFFICE SYMBOL (If applicable) AFWAL/FIGX	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB OH 45433	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
MULTIVARIABLE CONTROL LAW DESIGN FOR THE CONTROL RECONFIGURABLE COMBAT AIRCRAFT (CRCA)

12. PERSONAL AUTHOR(S)
Daryl Hammond, B.S.E.E., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Thesis	FROM ____ TO ____	1988 December	420

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Control Theory, Flight Control Systems, Multivariable Control, Adaptive Control Systems
01	04		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisor: Dr. John J. D'Azzo

Approved for public release in accordance with AFR 190-1
J. Benkiewicz
12 Jan 1989

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL John J. D'Azzo, Professor	22b. TELEPHONE (Include Area Code) (513) 255-3576 22c. OFFICE SYMBOL AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ABSTRACT

Typically, control law analysis and design for an aircraft include separating the longitudinal and lateral equations of motion and designing control laws for each separate motion. The simplifying assumptions are often valid and do not adversely affect the analysis and design when aerodynamic cross-coupling is minimal. The Control Reconfigurable Combat Aircraft (CRCA) design includes an all-flying canard with 30 degrees of dihedral angle which prevents the normal separation of lateral and longitudinal equations because of high aerodynamic cross-coupling. Consequently, developing a satisfactory controller for all aircraft motion must include all of the control surfaces and is more complicated.

The multivariable control law design used in this thesis incorporates the high-gain error-actuated Proportional plus Integral (PI) controller developed by Professor Brian Porter of the University of Salford, England. Control law development and simulation are performed using the computer aided design program called Xmat. Two successful fixed gain controller design methods and an adaptive controller design are demonstrated.

The three control surfaces on each wing are operated together, so they are treated in this thesis as one control effector. Thus, the five CRCA control inputs for this design consist of two canards, left trailing edge flaperon, right trailing edge flaperon, and rudder. The aircraft dynamics are linearized about three flight conditions. Fixed gain PI controllers are designed at each flight condition for both the healthy aircraft and with a failed left canard and left trailing edge flaperon. The simulation indicates that the controller is very robust and output responses are fully satisfactory.

An adaptive controller design, using a recursive least squares (RLS) parameter estimation algorithm, is developed for a self-tuning control system. When a left trailing edge flaperon failure occurs at the nominal flight condition, a new plant model is computed and the PI control law is revised accordingly. The ability of the controller to estimate the new plant parameters and compensate for the failed control surface is exceptional. Finally, a fixed gain Proportional plus Integral plus Derivative (PID) controller is designed for the nominal flight condition. The results are satisfactory. The design needs to be extended to the remaining flight conditions.